Pseudo-Random Trees Multiple Independent Sequence-Generators for Parallel and Branching Computations

TR88-037
July 1988
Revised-December 1988

John H. Halton

The University of North Carolina at Chapel Hill
Department of Computer Science
CB#3175, Sitterson Hall
Chapel Hill, NC 27599-3175

UNC is an Equal Opportunity/Affirmative Action Institution.
PSEUDO-RANDOM TREES
MULTIPLE INDEPENDENT SEQUENCE-GENERATORS
FOR PARALLEL AND BRANCHING COMPUTATIONS

by

John H. Halton
Computer Science Department
The University of North Carolina
Chapel Hill, NC 27599-3175, USA

ABSTRACT

A class of families of linear congruential pseudo-random sequences is defined, for which it is possible to branch at any event without changing the sequence of random numbers used in the original random walk, and for which the sequences in different branches show properties analogous to mutual statistical independence. This is a hitherto unavailable, and computationally desirable, tool.

1. INTRODUCTION

During the last forty or fifty years, the Monte Carlo method has been used with considerable success, to solve large mathematical problems too computationally complicated to yield to the classical numerical methods developed during the previous four centuries. For general discussions, the reader is referred to, e.g., BUS 62, HAM 64, HAL 70, ERM 71, SOB 73, KLE 75, YAK 77, or RUB 81 [references in this format are to the Bibliography at the end of this paper]. In particular, there is an extensive history of the effective application of the Monte Carlo method to particle-transport problems, such as arise in the design of radiation shielding, nuclear reactors, and fission and fusion bombs (see, e.g., SPA 69, CAR 75).

While the method was originally conceived in terms of representing the solution of a problem as a parameter of a hypothetical population, and using a [truly] random sequence of numbers to
construct a sample of the population, from which statistical estimates of the parameters can be obtained (see HAL 70); it soon became apparent, from the point of view of the need, both for repeatable results to 'debug' the Monte Carlo computer programs and for a large, stable supply of suitable 'random numbers', that certain deterministic sequences exhibiting some of the properties of truly random sequences would be more useful in practice. These became known as pseudo-random sequences (and, by corruption of terms, as sequences of 'pseudo-random numbers') (see the above-mentioned references, and also LEH 51, HUL 62, KNU 69, TAU 65, JAN 66, and NIE 78). Somewhat later, even less 'random-looking' sequences, dubbed quasi-random, having exceptionally good uniformity properties and leading to fast convergence of the resulting Monte Carlo estimates, were proposed (see HAM 60, HAL 60, and ZAR 68). The uniformity of distribution of the pseudo-random sequences was found to be imperfect when they were used to define points in several dimensions (FRA 63, GRE 65, MAR 72), and several non-statistical approaches were developed for error-analysis (HAL 60, ZAR 66, ZAR 68, HAL 72).

One of the most successful classes of pseudo-random number-generators is the so-called linear-congruential algorithm (originally due to Lehmer; see LEH 51). The sequence \[ \xi_0, \xi_1, \xi_2, \xi_3, \ldots \] \(\equiv\) \(\xi_j\)\(_{j=0}\) of canonical pseudo-random numbers, which should be independently uniformly distributed in the semi-open unit interval \([0, 1)\), is obtained from an integer sequence \(\{x_0, x_1, x_2, x_3, \ldots \} = \{x_j\}_{j=0}^{\infty}\) by

\[ \xi_j = x_j / 2^M; \quad (1) \]

and the \(x_j\) are uniquely determined by selecting \(M, a, b, \) and \(x_0\), and taking

\[ (\forall j \geq 0) \quad 0 \leq x_j < 2^M, \quad x_{j+1} \equiv ax_j + b \pmod{2^M}. \quad (2) \]

Given the integer parameters \(a\) and \(b\) and an initial integer \(x_0\), each successive \(x_{j+1}\) is the residue of \(ax_j + b\) modulo \(2^M\) (i.e., the remainder when \(ax_j + b\) is integer-divided by \(2^M\)). When we perform binary computations, such as are now universally used in digital computers, this residue is easily obtained, as the integer consisting of the \(M\) least significant bits of \(ax_j + b\). The value of \(M\) is mainly machine-dependent; in 'supercomputers', a typical value of \(M\) is 48, and then \(2^{48} \approx 2.8 \times 10^{14}\). Given integers \(Z\) and \(Q > 0\), we shall henceforth write

\[ R = \langle Z \rangle_Q \quad \leftrightarrow \quad 0 \leq R < Q, \quad R \equiv Z \pmod{Q}, \quad (3) \]
yielding \( (\forall j \geq 0) \quad x_{j+1} = \langle ax_j + b | 2^M \rangle \). \(^{(4)}\)

Many calculations using the Monte Carlo method (including those of particle transport alluded to above) involve the use of long sequences of pseudo-random numbers to generate sequential histories of flights and collisions, usually referred-to as random walks. By averaging appropriately-selected scores (functions of single random walks generated in this way) over large numbers of such random histories, it is possible to estimate the parameters of interest with considerable accuracy.

It is clear that different random sequences will, in general, produce different random-walk histories; and these latter, in turn, will generally lead to different scores. While it is inherent in the Monte Carlo method that its results should show random fluctuations, it is extremely convenient to be able to reproduce a given computational result exactly, when we wish to do so. In particular, this is important in the initial ‘debugging’ stage of developing a new program (or program-module), when we need to separate the effects of desirable randomness from those of undesirable programming errors, so as to ensure that the program or module will do correctly what the programmer intends; and it is also useful when several runs must be made, to develop intentionally-correlated random samples, all depending on the same random walk. Some of these ends can be achieved by storing, and later retrieving, the values of the thousands, millions, or even billions, of random numbers required; but it is clearly much more convenient to redesign the random generator (algorithm) in such a way that no such mass-storage is required. The original invention of pseudo-random sequences was partly motivated by this need.

When one attempts to refine the physics underlying a particle-transport computation, by taking into account the concomitant generation and subsequent motion of additional particles or radiation, it is useful to compare the scores obtained with and without these refinements, for the same random walks. Since this leads to situations in which the random walks branch in a tree-like manner, requiring random sequences of differing lengths and unpredictable relationships, the problem becomes far more complex. We are now required to be able to generate a tree-structure of pseudo-random numbers, with good uniformity properties within each branch and good properties of independence between branches. In a typical conventional particle-transport calculation, using non-branching random walks, we may compute some \(10^3 - 10^8\) random walks, averaging perhaps \(10^2 - 10^6\) steps each, with every step requiring around 10 random numbers; this adds up to a need for something of the order of \(10^6 - 10^{15}\) random numbers. With current generators
having periods of the order of $10^{14}$, such a requirement is acceptable; since techniques are available to increase the periods (without unacceptably increasing the time required to generate the random numbers) to the order of $10^{60}$ or so.

However, if our model is expanded to allow branching at every step, a comparable tree-structured calculation would, in principle, need some $10^4 \times 2^{10^2} = 10^{34}$ to $10^9 \times 2^{10^6} = 10^{301039}$ random numbers. It is, of course, entirely out of the question, in any case, to use this many random numbers; since, according to current astrophysical thought, the calculation would hardly have begun when the Sun, in its red-giant phase, would consume the Earth, just a mere $10^{26} - 10^{27}$ nanoseconds from now! The problem is, rather, to provide theoretical access to suitably-distributed random numbers; so that they will be available as and when needed. The actual consumption of random numbers in a computation of this kind could hardly exceed some $10^{16}$ or so, unless computer technology makes rather remarkable progress even in comparison with its astonishing record; thus, we must rely on sampling techniques such as 'Russian roulette' to keep the overall needs down. Nevertheless, we must be able to generate those random numbers that we do need, with appropriate properties of distribution. The present development is an attempt to address this potential need. The problem was first raised by Warnock (see war 83) and useful suggestions of a general and heuristic nature were made by him as to its solution. In the present paper, I propose a possible explicit approach to the task of generating a large number of branching pseudo-random sequences which are mutually independent in a rigorously specified manner.

2. PRELIMINARIES

For any positive integer $n$ and real $a$, let

$$S_0(a) = 0 \quad \text{and} \quad S_n(a) = 1 + a + a^2 + a^3 + \ldots + a^{n-1}. \quad (5)$$

This is consistent, since the sum $S_n(a)$ has $n$ terms. Then

$$S_n(a) = n, \quad \text{if} \quad a = 1, \quad (6)$$

and

$$S_n(a) = (a^n - 1)/(a - 1), \quad \text{if} \quad a \neq 1. \quad (7)$$

**Lemma 1.** For any non-negative integer $m$ and real $z$,

$$S_{2m}(z) = (1 + z) S_m(z^2). \quad (8)$$
By (5), if \( m = 0 \), then (8) is immediate; and, otherwise,

\[
S_{2m}(z) = (1 + z) + (z^2 + z^3) + \ldots + (z^{2m-2} + z^{2m-1})
= (1 + z) (1 + z^2 + z^4 + \ldots + z^{2m-2});
\]

which yields (8) at once. [Proofs will, throughout this paper, be enclosed between \(<\langle \text{ and } \rangle\>\).

**Definition 1.** If \( N \) is any positive integer, then we express the fact that another positive integer \( k \) is a factor of \( N \) [i.e., integer divides it, without remainder] by the usual notation

\[
k \mid N.
\]

We now see, in particular, that there is a unique non-negative integer \( u \) such that \( k^u \) divides \( N \), but \( k^{u+1} \) does not. We shall write

\[
k^u \nmid N
\]

to express this situation. If \( v \leq u \), then we also have, as in (10), that

\[
k^v \mid N.
\]

We extend the notation (11) to \( N = 0 \) by writing, for any \( k > 0 \),

\[
k^\infty \nmid 0.
\]

The notation defined in (11) and (13) is slightly tricky: while \( k \mid N \) is a relation between two integers, \( k \) and \( N \); \( k^u \nmid N \) is a relation between three integers, \( k \), \( u \), and \( N \). When we use an abbreviation, such as \( "8 \nmid x " \), it will be understood to mean \( "2^3 \nmid x " \): the member on the left of the symbol \( \nmid \) will always be a pure power of one uniquely determined \( k \). Hereinafter, we shall particularly make use of the special case, when \( k = 2 \).

**Lemma 2.** For any odd positive integer \( a \), there are unique positive integers \( q \) and \( r \), such that

\[
a = (2r - 1) 2^q - 1.
\]

Since \( a \) is odd, \( a + 1 \) is necessarily even. Thus, there is a unique maximum \( q \) for which \( 2^q \mid (a + 1) \), and \( q \geq 1 \). For this \( q \), we have \( 2^q \nmid (a + 1) \). Also, the quotient, when we divide \( a + 1 \) by \( 2^q \), is odd; whence it can be expressed uniquely in the form \((2r - 1)\). This immediately yields (14).
Lemma 3. With $a$, $q$, and $r$ defined as in Lemma 2; if $u \geq 0$ and $v \geq 0$ are the unique integers such that $2^u \nmid n$ and $2^v \nmid S_n(a)$, then $v = u + q - 1$; that is,

$$2^{u+q-1} \nmid S_n(a) \quad \text{if and only if} \quad 2^u \nmid n. \quad (15)$$

By repeated application of Lemma 1, we get that

$$S_n(a) = (1 + a) S_{n/2}(a^2) = (1 + a) (1 + a^2) S_{n/4}(a^4) = \ldots = (1 + a) (1 + a^2) (1 + a^4) \ldots (1 + a^{2^{u-1}}) S_{n/2^u}(a^{2^u}). \quad (16)$$

Also, by (14), $2^q \nmid (1 + a)$, and $q \geq 1$; and every binomial factor on the right of (16), after the first one, is of the form $1 + a^{2m}$, with integer $m \geq 1$. Since $a$ is odd, either $a = 1$ or $a = 3 \pmod{4}$; whence $a^2 = 1 \pmod{4}$; and, therefore,

$$(\forall m \geq 1) \quad a^{2m} = 1 \pmod{4}. \quad (17)$$

Hence, $a^{2m} \equiv 2 \pmod{4}$; i.e., $(\forall m \geq 1) 2 \nmid (1 + a^{2m})$. Therefore, the product of all the binomial factors on the right of (16) is divisible by $2$ exactly $q + (u - 1)$ times. Finally, we observe that, since $a$ is odd by our hypothesis, every power of $a$ is odd too; whence, by (5), the last factor on the right of (16) is the sum of an odd number, $n/2^u$, of odd numbers, and so must itself be odd. Thus, when $u$ and $v$ are defined as stated, $v = q + u - 1$, and (15) follows immediately. $\gg$

**Definition 2.** If $[x_0, x_1, x_2, \ldots] = [x_j]_{j=0}^\infty$ is a sequence of numbers, and if we are given that, for some $0 \leq i < j$,

$$(\forall k \geq 0) \quad x_{j+k} = x_{i+k}. \quad (18)$$

then we say that the sequence is periodic. If $\lambda$ is the least value of the difference $j - i$, for which (18) holds, then we say that the period is $\lambda$.

If $h$ is the least value of $i$ satisfying (18) for $j - i = \lambda$, we say that the periodicity starts at Index $h$; and if $h = 0$, then we say that the sequence is completely periodic.

Note that, if the sequence $[x_j]_{j=0}^\infty$ is periodic, with period $\lambda$, starting at index $h$; then, for any offset $\alpha$, the same is true of the sequence $[x_j - \alpha]_{j=0}^\infty$. 

---
Lemma 4. Given that the sequence \(\{x_j\}_{j=0}^{\infty}\) is periodic with period \(\lambda\), starting at index \(h\), and given \(i\) and \(j\), with \(i < j\), satisfying the relation (18); it follows that \(\mu = j - i\) is an integer multiple of \(\lambda\); that is,

\[
\lambda \mid \mu. \quad (19)
\]

Since \(\lambda\) is minimal, we have \(0 < \lambda \leq \mu\). Because the sequence is periodic with period \(\lambda\), starting at index \(h\); it is clear from (18) that \(x_{h+k} = x_{(h+\lambda)+k} = x_{(h+\lambda)+(\lambda+k)} = x_{h+(2\lambda+k)} = \ldots\); that is, by induction on integers \(r\),

\[
(\forall k \geq 0) (\forall r \geq 0) \quad x_{h+r\lambda+k} = x_{h+k}; \quad (20)
\]

and, similarly, by (18) for \(i\) and \(j\), by induction on integers \(s\),

\[
(\forall k \geq 0) (\forall s \geq 0) \quad x_{i+su+k} = x_{i+k}. \quad (21)
\]

Write \(n = \max(i, h)\), so that \(n \geq h\) and \(n \geq i\); and replace \(k\), throughout (20), by \(k + n - h\) and, throughout (21), by \(k + n - i\). Then, whatever is true with the resulting universal quantifiers, namely, \((\forall k \geq h - n)\) and \((\forall k \geq i - n)\), is also true with the quantifier \((\forall k \geq 0)\); so that

\[
(\forall k \geq 0) (\forall r \geq 0) (\forall s \geq 0) \quad x_{n+r\lambda+k} = x_{n+k} = x_{n+su+k}. \quad (22)
\]

The Euclidean Algorithm Theorem states that, if \(\gamma\) denotes the g.c.d. of positive \(\lambda\) and \(\mu\) (so that \(\gamma \mid \lambda\) and \(\gamma \mid \mu\), and \(\gamma\) is maximal), there are integers \(U_0\) and \(V_0\) such that \(\gamma = U_0\lambda + V_0\mu\).

Proof: Let \(Z\) be the set of all integers. The set \(\Theta = \{\theta = U\lambda + V\mu: U \in \mathbb{Z}, V \in \mathbb{Z}, \theta > 0\}\), which is non-empty, since \(0 < \lambda = 1 \times \lambda + 0 \times \mu \in \Theta^+\) and \(0 < \mu = 0 \times \lambda + 1 \times \mu \in \Theta^+\). Let \(\kappa = U_0\lambda + V_0\mu\) be the least \(\theta \in \Theta^+\). Integer-divide \(\lambda\) by \(\kappa\); then \(\lambda = \sigma \kappa + \rho\) (where \(0 \leq \rho < \kappa\)), and so \(\rho = \lambda - \sigma \kappa = (1 - \sigma U_0)\lambda - \sigma V_0\mu \in \Theta\). Since \(\rho < \kappa\), and \(\kappa\) is minimal in \(\Theta^+\), \(\rho \in \Theta^+\); and therefore \(\rho = 0\) (i.e., \(\kappa \mid \lambda\)). Integer-divide \(\mu\) by \(\kappa\), to show, similarly, that \(\kappa \mid \mu\); whence \(\kappa \mid \gamma\), since \(\gamma\) is the maximal divisor. Since we also know that \(\gamma \mid \lambda\), \(\gamma \mid \mu\), and \(\kappa \in \Theta\); \(\gamma \mid \kappa\). Therefore, \(\kappa = \gamma\). This proves the theorem. Now, \(U_0\) and \(V_0\) must have opposite signs, since we have that \(0 < \gamma \leq \lambda \leq \mu\); so that there must be non-negative integers \(r_0\) and \(s_0\), such that either (i) \(r_0\lambda - s_0\mu = \gamma\) or (ii) \(s_0\mu - r_0\lambda = \gamma\). In both cases, take \(r = r_0\) and \(s = s_0\).
in (22); then, in case (i), replace \( n \) by \( v - s_0 \mu \); in case (ii), replace \( n \) by \( v - r_0 \lambda \). Either way, we see that

\[
(\forall k \geq 0) \quad x_{v+k} = x_{v+k}.
\]  

But this means that the sequence is periodic, with period at most \( \gamma \). Since \( \lambda \) is minimal, by Definition 2, we must have \( \lambda \leq \gamma \). Thus, \( \gamma = \lambda \), and the lemma follows at once.>

This means that the period of a periodic sequence is unique.

**Definition 3.** Given a semi-open interval \([A, B]\) on the real line, and a set \( J \) of \( Q \) points \( z_1 < z_2 < \ldots < z_Q \) in it, we say that the points are cyclically equally spaced (CES) in \([A, B]\) if

\[
z_{h+1} - z_h = (B - A)/Q \quad \text{for} \quad h = 1, 2, \ldots, Q - 1.
\]  

Note that this implies that \( (z_1 - A) + (B - z_Q) = (B - A)/Q \) also, since \( z_Q - z_1 = (Q - 1)(B - A)/Q \). If we imagine the interval \([A, B]\), with the points of \( J \) in it, wrapped around a circle; then these \( Q \) points would be equally-spaced around the circle. Note, too, that, if the set \( J \) is CES in \([A, B]\), so is any offset set of points \( z_h - \alpha \) (reduced, modulo \( B - A \), to fall in the interval).

Though it is not necessary to do so, we hereafter limit ourselves to integer sequences and the interval \([0, Q]\).

**Definition 4.** Given the set \( J = \{0, 1, 2, \ldots, Q - 1\} \), CES in \([0, Q]\); if the sequence \( \{x_j\}_{j=0}^{\infty} \) is periodic, with period \( \lambda \), starting at index \( h \), and if the set \( K_0 = \{x_j\}_{j=h}^{\infty} \) of values taken by the \( x_j \), once the periodicity is established, is a subset of \( J \), with \( P \) distinct points in it; and if, further, these \( P \) values are also CES in \([0, Q]\), and \( P = \lambda \); then we say that the sequence is uniform in \( J \), with coarseness \( Q/P \).

**Lemma 5.** In the situation described in Definition 4,

\[
P \mid Q:
\]  

so that the coarseness of a uniform sequence is always a positive integer.

<<The points of \( J \) may be thought of as equally spaced around a circle of circumference \( B - A \); the points of \( K \) (which are also in \( J \)) are also equally spaced around the circle. Thus, there is an integer \( G \), such that adjacent points of \( K \) have a spacing just \( G \) times as great as the circle radius.>>
that of adjacent points of \( J \); that is, \( PG = Q \); whence (25) follows. \( G \) is therefore the coarseness of the sequence in \( J \).

Note that, if the period of the sequence \( [x_j]_{j=0}^\infty \) passes through all the points of \( J \) (that is, if \( P = Q \)), then the coarseness of the sequence in \( J \) takes its minimum possible value, namely, 1.

**Definition 5.** Given the set \( J = \{0, 1, 2, \ldots, Q - 1\} \), CES in \( [0, Q) \): if two sequences \( [x_j]_{j=0}^\infty \) and \( [x'_j]_{j=0}^\infty \) are such that the difference-sequence, \( [\delta_j]_{j=0}^\infty \), where

\[
(\forall j \geq 0) \quad \delta_j = \langle x_j - x'_j \rangle Q.
\]

is periodic, and is uniform in \( J \) with coarseness \( G \); then we say, by analogy with the definition of uniformity and coarseness, that the two sequences are *independent* with respect to \( J \), and that their *consonance* is \( G \).

### 3. ANALYSIS OF LINEAR CONGRUENTIAL GENERATORS

We are interested in generating a canonical pseudo-random sequence \( [\xi_j]_{j=0}^\infty \) of numbers in \( [0, 1) \), for use in Monte Carlo computations. We therefore want the \( \xi_j \) to take a large number of distinct values, distributed with near-constant density in \( [0, 1) \). Our present consideration will be limited to the linear congruential sequences, which are related through (1) to the integer sequences \( [x_j]_{j=0}^\infty \) defined in (2) or (4), with \( M \) a non-negative integer. This implies that, if we write (as we shall do henceforth)

\[
2^M = Q,
\]

then

\[
(\forall j \geq 0) \quad x_j \in J = \{0, 1, 2, \ldots, Q - 1\},
\]

and therefore

\[
(\forall j \geq 0) \quad \xi_j \in F = \{0, 1/Q, 2/Q, \ldots, (Q - 1)/Q\}.
\]

In the terminology of Definition 3, the sets \( J \) and \( F \) are CES, in the semi-open intervals \( [0, Q) \) and \( [0, 1) \), respectively.
Note that we may (and do, henceforth) assume, without loss of
generality, that $a$ and $b$ are also integers selected from $J$. We further
assume that $a \neq 0$. [If $a = 0$, then, clearly, by (4), for all $j \geq 1$, $x_j = b$.]

**Lemma 6.** The recurrence relation (4) is satisfied, for all $n \geq 0$,
by

$$x_n = \langle a^n x_0 + S_n(a) b \mid \mathcal{G} \rangle;$$

(30)

where $S_n(a)$ is defined as the sum in (5).

<<When $n = 0$, we know that $a^n = 1$ and the sum $S_n(a) = 0$; so
that, in fact, $x_n = a^n x_0 + S_n(a) b$. Suppose that the relation holds for
$n = k$, say (this is initially true when $k = 0$). Then, by (4) with (3), we
have that

$$x_{k+1} = \langle a x_k + b \mid \mathcal{G} \rangle = \langle a [a^k x_0 + S_k(a) b] + b \mid \mathcal{G} \rangle$$

$$= \langle a^{k+1} x_0 + [a S_k(a) + 1] b \mid \mathcal{G} \rangle;$$

(31)

and, by (5), it is easily seen that

$$S_k(a) + 1 = S_{k+1}(a);$$

whence the congruence will also hold for $n = k + 1$. The lemma
follows by induction.>>

**Lemma 7.** The sequence $[x_j]_{j=0}^{\infty}$ is periodic, with period not
exceeding $Q$.

<<By (28), there are at most $Q$ possible distinct values of $x_j$;
among the $Q + 1$ numbers $x_0, x_1, x_2, \ldots, x_Q$, there must be two values
alike, and we can always further specify that all intermediate values
different from these and each-other: $x_i = x_j$, say, with $0 \leq i < j$ and
$x_i, x_{i+1}, x_{i+2}, \ldots, x_{j-1}$ all different [if some intermediate value
$x_k = x_i$, say, replace $j$ by $k$; if two intermediate values $x_n = x_k$, say,
replace $i$ by $h$ and $j$ by $k$]. It is now clear from the form of (4) that (18)
will hold, since each member of the sequence is determined solely
and uniquely by its immediate predecessor, without regard to its
position in the sequence. Hence, the sequence is periodic and, by
Lemma 4, $j - i$ is a multiple of the period, which thus, clearly, cannot
exceed $Q$.>>
Lemma 8. If \( a \) is any even integer, the sequence \([x_j]_{j=0}^{\infty}\) is periodic, with the period 1.

We have already seen that the period is 1 when \( a = 0 \). For any even \( a \), clearly \( a^M = 0 \) (mod \( Q \)); so there will be a unique minimal \( h \), such that \( a^h = 0 \) (mod \( Q \)). If \( n \geq h \); then, by (5),

\[
S_n(a) = S_h(a) + a^h S_{n-h}(a) \equiv S_h(a) \pmod{Q}. \quad (33)
\]

Therefore, in particular, by (30) and (33),

\[
x_{h+1} = \left< a^{h+1} x_0 + S_{h+1}(a) \right| b \pmod{Q} = \left< S_h(a) \right| b \pmod{Q}
\]

\[
= \left< a^h x_0 + S_h(a) \right| b \pmod{Q} = x_h \quad (34)
\]

whence, by Definition 2, the sequence is periodic, starting at index \( h \), with period 1.>

Of course, a period of length 1 is of very little use for the generation of pseudo-random numbers; so we shall henceforth assume that \( a \) is odd.

Lemma 9. If \( a \) is any odd integer, then the sequence \([x_j]_{j=0}^{\infty}\) is completely periodic.

Consider the \( Q \) integers \( 1, a, a^2, \ldots, a^Q \), reduced modulo \( Q \). Their values must lie in the set \( J \); so, arguing exactly as in proving Lemma 7, we see that we must have \( 0 \leq i < j \leq Q \), such that \( <a^i|Q> = <a^j|Q> \), while \( <a^i|Q>, <a^{i-1}|Q>, <a^{i-2}|Q>, \ldots, <a^{-1}|Q> \) are all different. Thus, \( a^j - a^i = a^i(a^{j-i} - 1) \) must be divisible by \( Q \); and since \( a \) is odd, it follows that \( Q \mid (a^{j-i} - 1) \); so that there must be a positive integer \( m = j - i \leq Q \), such that

\[
a^m = 1 \pmod{Q}. \quad (35)
\]

By (2) and (35), we have that \( x_{j-1} = a^m x_{j-1} = a^{m-1}(x_j - b) \pmod{Q} \); so that, writing \( c = a^{m-1} \) and \( d = -cb \), we have

\[
(\forall j \geq 1) \quad x_{j-1} \equiv cx_j + d \pmod{Q}. \quad (36)
\]

or, by (3),

\[
(\forall j \geq 1) \quad x_{j-1} = \left< cx_j + d \right| Q>. \quad (37)
\]
Thus, each member of the sequence is determined solely and uniquely by its immediate successor, without regard to its position in the sequence, and the equation (18) also holds for negative $k$, so long as the index $i + k \geq 0$. This extends the periodicity of the sequence (already established in Lemma 7) to the starting index 0, proving the present lemma.>>

From now on, we shall always suppose that $a$ is odd, satisfying (14) and thereby uniquely defining positive integers $q$ and $r$, as stated in Lemma 2. Since we also suppose (without loss of generality) that $a \in J$, we see, by (28), that $1 \leq (2r - 1) 2^q - 1 \leq 2^M - 1$; whence $r \geq 1$, and therefore $2^q \leq 2^M$. Since $q \geq 1$, we conclude that

$$1 \leq q \leq M. \tag{38}$$

Now write $W = \langle x_1 - x_0 | Q \rangle = \langle (a - 1)x_0 + b | Q \rangle$;

and, by appeal to Definition 1, put

$$2^c \uparrow b, \quad 2^s \uparrow x_0, \quad 2^d \uparrow (a - 1), \quad \text{and} \quad 2^g \uparrow W. \tag{40}$$

Since (again without loss of generality) we also suppose that $b \in J$ and $x_0 \in J$, it now follows that, unless $b = 0$ [$c = \infty$] or $x_0 = 0$ [$s = \infty$],

$$0 \leq c < M \quad \text{and} \quad 0 \leq s < M; \tag{41}$$

and, since $a$ is odd, $a - 1$ is even, whence $d \geq 1$.

**Lemma 10.** The period $\lambda$ of the completely periodic sequence $[x_j]_{j=0}^\infty$ is given by

$$\lambda = 2^u, \quad \text{where} \quad u = \max(0, M - g - q + 1), \tag{42}$$

and $g$ is defined uniquely by (39) and (40).

<<By Definition 2 and (30), $\lambda$ is the least $j$ for which

$$x_0 = x_j = \langle a'x_0 + S_j(a) b | Q \rangle. \tag{43}$$

If $a \neq 1$, by (7), $a'x_0 - x_0 = S_j(a) (a - 1) x_0$; whence, by (3), (39), and (43),

$$S_j(a) W \equiv 0 \pmod{Q}. \tag{44}$$

If $a = 1$, we note that $W = b$, and so (43) implies (44) directly. Thus (44) is true for all $a$. Therefore, either
Let \( W = 0 \pmod{Q} \) (mod \( Q \)) (45)

(i.e., \( g \geq M \), including the possibility that \( W = 0 \) and \( g = \infty \); or \( g < M \), and

\[ 2^{M-g} \mid S_j(a). \] (46)

If (45) holds, then clearly, by (4) and (39), \( x_1 = x_0 \); so that \( \lambda = 1 \).

Thus, \( u = 0 \) and \( M - g - q + 1 \leq 0 \) [since, by the assumption of (45), \( g \geq M \). and, by (38), \( q \geq 1 \)]; so that (42) is satisfied.

If, instead, \( g < M \) and (46) holds, we observe that, by Lemma 3,

\[ 2^{u+q-1} \mid S_j(a) \] if and only if \( 2^u \mid j \); whence there is an integer \( u \geq 0 \), such that \( u + q - 1 \geq M - g \) and \( 2^u \mid \lambda \). Thus, since the period \( \lambda \) is minimal, \( u \) will be the least non-negative solution of

\[ \lambda = 2^u \quad \text{and} \quad u + q - 1 \geq M - g. \] (47)

Clearly, this is given by (42).

**Lemma 11.** With \( g \) defined by (39) and (40);

(i) if \( c < s + d \), then \( g = c \);

(ii) if \( c = s + d \), then \( g > c \);

(iii) if \( c > s + d \), then \( g = s + d \).

By (40), \( 2^{s+d} \mid (a - 1)x_0 \) and \( 2^c \mid b \). Write \((a - 1)x_0 = 2^{s+d} U \) and \( b = 2^c V \), where \( U \) and \( V \) are odd integers. By (39), there are now three cases, characterized as in our lemma. (i) If \( c < s + d \), then \( W = \langle (a - 1)x_0 + b \mid Q \rangle = \langle 2^c(2^{s+d-c} U + V) \mid Q \rangle = 2^c X_1 \), and the factor \( X_1 \) is odd; so that \( g = c \). (ii) If \( c = s + d \), then \( W = \langle 2^c(U + V) \mid Q \rangle = 2^c X_2 \), and the factor \( X_2 \) is even, being the sum of two odd numbers; so that \( 2^{c+1} \mid W \) (that is, \( g > c \)). (iii) If \( c > s + d \), then \( W = \langle 2^{s+d} [U + 2^{c-s-d} V] \mid Q \rangle = 2^{s+d} X_3 \), and the factor \( X_3 \) is odd; so that \( g = s + d \).

As we shall see later, it is not always possible to control the parity of \( b \); but we can, and do, control the value of \( a \) (and thus the parity of \( a - 1 \)). We naturally seek to make the period of the sequence as long as possible. The absolute maximum is clearly \( Q = 2^M \), but this cannot always be attained. Referring to Lemma 10, we see that both \( q \) and \( g \) should be as small as possible; and, since, by (36), \( q \geq 1 \), we stipulate that
By the definition (14) of \( q \) and \( r \), this is equivalent to
\[
a = (2r-1)2 - 1 = 4(r-1) + 1; \quad \text{so that}
\]
\[
a = 1 \pmod{4}. \quad (49)
\]
By the definition (40) of \( d \), we have that, for some integer \( r' \),
\[
a = (2r' - 1)2^d + 1 \quad (50)
\]
[compare (14)], which implies that
\[
a = 1 \pmod{2^d}. \quad (51)
\]
Now, we have (above) that \( a - 1 = 4(r-1); \quad \text{so that, by (50),}
\]
\[
d \geq 2. \quad (52)
\]
Conversely, by (50), if we assume (52), \( a - 1 = (2r - 1)2^d = 4r' \), which implies (49); further, \( a = (2r'' + 1)2 - 1 \), which yields (48), by (14).

First, let us consider what happens when \( b \neq 0 \).

**Lemma 12.** Under the conditions of Lemmas 10 and 11, if we impose the restrictions (50) and (52) on the parameter \( a \) and suppose that \( b \neq 0 \), then

(i) if \( c \leq s + d - 1 \), the period of the sequence is \( 2^{M-c} \geq 2; \)

(ii) if \( c = s + d \), the period of the sequence is \( \max[1, 2^{M-g}]; \) where \( g \geq c + 1; \)

(iii) if \( c \geq s + d + 1 \), the period of the sequence is \( 2^{M-s-d} \geq 4.\)

Without regard to \( b \), we know that (50) and (52) imply \( q = 1.\)

Thus, (42) reduces to
\[
\lambda = 2^u, \quad \text{where} \quad u = \max[0, M-g]; \quad (53)
\]
and the three cases of Lemma 11 are the same as those of the present lemma. Now restrict consideration to \( b \neq 0 \).

(i) If \( c \leq s + d - 1 \), then \( g = c \). By (41), since \( b \neq 0, \ c < M, \) and it follows that \( M - g = M - c \geq 1; \) so that, by (53), \( \lambda = 2^{M-c} \geq 2^1 = 2.\)

(ii) If \( c = s + d \), then \( g > c; \) and, by (53), \( \lambda = \max[1, 2^{M-g}].\)

(iii) If \( c \geq s + d + 1 \), then \( g = s + d \). Since \( b \neq 0, \) by (41) and our hypothesis, \( s + d < c < M, \) so we get that \( M - g = M - s - d \geq 2; \) so that, by (53), \( \lambda = 2^{M-s-d} \geq 2^2 = 4.\)
Now we turn to the omitted case, when \( b = 0 \) and \( c = \infty \). By (4) or (30), we see that

\[
x_n = \ll a^n x_0 | G \rr.
\]  

(54)

Therefore, if \( x_0 = 0 \), every \( x_n = 0 \) too; so that \( \lambda = 1 \). If, on the other hand, \( x_0 \neq 0 \), so that \( 2^s \uparrow x_0 \), with \( 0 \leq s < M \); we can write \( x_0 = 2^s \omega_0 \), where \( \omega_0 \) is odd, and we see that (since \( a \) is odd) \( 2^s \uparrow x_n \) too; so that, for all \( n \),

\[
x_n = 2^s \omega_n,
\]

(55)

where \( \omega_n \) is odd. Thus, (54) reduces, on division by \( 2^s \), to

\[
\omega_n = \ll a^n \omega_0 | 2^{M-s} \rr.
\]  

(56)

We are therefore led to examine the dependence on \( m = M - s \) of the period \( \lambda_m \) of the sequence \( \{\omega_j\}_{j=0}^{\infty} \) with \( \omega_0 \) (and therefore all the \( \omega_j \)) odd, when all numbers are reduced modulo \( 2^m \). By (56), this problem is seen to be equivalent to that of finding the least \( n \) for which

\[
a^n \equiv 1 \pmod{2^m}.
\]  

(57)

By (50) and (52), and since, clearly, if \( u \geq v \),

\[
X = Y \pmod{2^u} \Rightarrow X = Y \pmod{2^v};
\]

(58)

it follows that the \( \lambda_m \) are nondecreasing as \( m \to \infty \), and that

\[
\lambda_1 = \lambda_2 = \ldots = \lambda_d = 1.
\]

(59)

As a further preliminary, we need the following result.

**Lemma 13.** When \( a \) satisfies (50) and (52), the least value of \( n \) for which (57) holds is \( 2^{m-d} \), for all \( m \geq d \).
Since $\lambda_m$ is the least $n$ for which (57) holds; for each $m$, there is an integer $q_m$ such that
\[
a^{\lambda_m} = 1 + q_m 2^m.
\] (60)
Suppose it known that $\lambda_m = 2^m - a$ for all $d < m < h$; by (59), this is certainly true for $h = d$. Putting $\lambda_h = 2^h - a$ in (60), we get that
\[
a^{2^h - a} = 1 + q_h 2^h; \text{ and, on squaring, this yields}
\]
\[
a^{2^{h+1} - a} = \left(a^{2^h}ight)^2 = \left(1 + q_h 2^h\right)^2 = 1 + q_h 2^{h+1} + q_h^2 2^{2h}.
\]
Therefore, since $h \geq d \geq 2$, by (52), we get that
\[
a^{2^{h+1} - a} = 1 \pmod{2^{h+1}};
\] (61)
whence $\lambda_{h+1} \leq 2^{h+1} - a$. Further, since the $\lambda_m$ are nondecreasing, we get $\lambda_{h+1} \geq \lambda_h = 2^h - a$. If we let $X = \lambda_h - 2^h - a$, so that $0 \leq X \leq 2^h - a$, then
\[
a^{\lambda_h + 1} = a^{X - 2^h - a} = a^X a^{2^h - a} = a^X (1 + q_h 2^h).
\] (62)
Let $a^X = Y + s 2^h$, with $0 \leq Y < 2^h$. Then $a^{\lambda_{h+1}} = (Y + s 2^h) (1 + q_h 2^h) = Y + (Y q_h + s) 2^h = Y + Z 2^h \pmod{2^{h+1}}$, where $Z = \langle Y q_h + s \rangle 2$ is
\[
0 \text{ or } 1. \text{ Since } a^{\lambda_{h+1}} = 1 \pmod{2^{h+1}} \text{ and } 0 \leq Y < 2^h, \text{ it is clearly necessary that } Y = 1 \text{ and } Z = 0; \text{ so that } a^X = 1 \pmod{2^h}; \text{ whence }
\]
\[
X \geq 2^h - a. \text{ Since we also have } X \leq 2^h - a, \text{ it follows that } X = 2^h - a; \text{ whence }
\]
\[
\lambda_{h+1} = 2^h - a + X = 2^h - a + 2^h - a = 2^{h+1} - a. \text{ The lemma now follows by induction.} \]

**Lemma 14.** When $a$ satisfies (50) and (52) and $b = 0$, the period of the sequence $[x_j]_{j=0}^\infty$ is $\max\{1, 2^{M-s-d}\}$.

\(<\langle i \rangle \text{ If } x_0 = 0, s = \infty \text{ and, as we have seen, } \lambda = 1, \text{ agreeing with the lemma.} \rangle \text{ ii) If } x_0 \neq 0 \text{ and } M - s - d \leq 0; \text{ then } 1 \leq M - s \leq d, \text{ by (41).} \text{ Since } m = M - s \text{ in (57), we get by (59) that } \lambda = \lambda_{M-s} = 1, \text{ again agreeing with the lemma.} \rangle \text{ iii) Otherwise, } x_0 \neq 0 \text{ and } M - s - d > 0; \text{ and the lemma asserts that the sequence } [x_j]_{j=0}^\infty \text{ has a period } 2^{M-s-d}. \text{ Now, the period of the sequence } [x_j]_{j=0}^\infty, \text{ given by (54), is clearly, by}
(55) and (56), the same as that of the sequence \([\omega_j]_{j=0}^\infty\) with \(\omega_0\) odd; and this, in turn, equals the least \(n\) for which (57) holds, when \(m = M - s\). By Lemma 13, this is \(2^{M-s-d}\), completing the proof of our lemma.>>

Lemmas 12 and 14 show the general desirability of using odd values of \(b\). Then, \(c = 0\), and we are in Case (i) of Lemma 12, with \(\lambda = 2^M\), the optimal situation. However, as we shall see later, this will not always be possible to achieve.

It is interesting to see under what circumstances the least desirable situation (namely, when \(\lambda = 1\)) occurs. We already know, by Lemma 8, that this can happen when \(a\) is even. Lemma 12 now tells us that, when \(a\) is odd and satisfies (50) and (52), and \(b \neq 0\), it can only happen in Case (ii), when \(c = s + d\). Let us write

\[
x_0 = 2^M - \theta, \quad a - 1 = 2^M - a, \quad b = 2^M - \beta;
\]

where, by (50), \(a = 2^d \ (2U - 1)\) with \(1 \leq U \leq 2^{M-d-1}\); and, since \(b\) and \(x_0\) are in \(J\), \(\beta = 2^{s+d} \ (2V - 1)\) with \(1 \leq V \leq 2^{M-s-d-1}\), and \(\theta = 2^s \ (2X - 1)\) with \(1 \leq X \leq 2^{M-s-1}\). Then, by (39),

\[
W = \langle 2^{2M} - 2^M (a + \theta - 1) + \alpha\theta - \beta \rangle, \quad 0
\]

and therefore, by (53), we get that \(\lambda = 1\) if and only if \(g \geq M\); i.e., if and only if

\[
\beta \equiv \alpha\theta \pmod{Q}, \quad \text{i.e.} \quad V \equiv 2UX - U - X + 1 \pmod{2^{M-s-d-1}}. \quad (65)
\]

Finally, Lemma 14 tells us that we can have \(\lambda = 1\) when \(b = 0\), either if \(x_0 = 0\) or if \(x_0\) is a multiple of \(2^M-d\).

**Lemma 15.** If the sequence \([x_j]_{j=0}^\infty\) is generated by (4), with the parameter \(a\) odd, then, given (40), we have that

(a) if \(c < \infty\) and \(c \leq s - 1\), \((\forall j \geq 0) \ \{2^{c+1} \mid x_{2j} \text{ and } 2^c \uparrow x_{2j+1}\};

(b) if \(c = s < \infty\), \((\forall j \geq 0) \ \{2^c \uparrow x_{2j} \text{ and } 2^{c+1} \mid x_{2j+1}\};

(c) if \(c \geq s + 1\) or if \(c = s = \infty\), \((\forall j \geq 0) \ 2^s \uparrow x_j.\)
For all \( j \geq 0 \), define the powers \( t_j \) by
\[
2^j = x_j. \tag{66}
\]
Then, by (4), since \( a \) is odd, \( 2^j \parallel ax_j \), and, by (40), \( 2^c \parallel b \). We recall that it is possible for \( b \) or any \( x_j \) to vanish, yielding that \( c = \infty \) or \( t_j = \infty \), respectively [see (13)]. Using an argument exactly analogous to that used in proving Lemma 11, we see that (i) if \( b = x_j = 0 \), then \( c = t_j = \infty \), and, in fact, every \( x_n = 0 \) [including \( x_0 = 0 \); see (2) and (36)]; so that \( s = \infty \) and \((\forall j \geq 0) \ 2^c \parallel x_j \); (ii) if \( c < t_j \), then \( 2^c \parallel x_{j+1} \); (iii) if \( c = t_j \), then \( x_{j+1} \) must be an even multiple of \( 2^c \), so that \( 2^{c+1} \parallel x_{j+1} \); and (iv) if \( c > t_j \), then \( 2^b \parallel x_{j+1} \). Thus,
\[
t_j > c \implies t_{j+1} = c; \quad t_j = c \implies t_{j+1} > c; \quad t_j < c \implies t_{j+1} = t_j. \tag{67}
\]
But the sequence \([t_j]_{j=0}^{\infty} \) begins with \( t_0 = s \); whence the lemma follows immediately.>>

Lemma 16. Given \( M > 0 \), with \( Q = 2^M \) and \( L = [0, Q] \); define the set \( J \) by (28), and let the sequence \([x_j]_{j=0}^{\infty} \) be periodic, with period \( \lambda \), starting at index \( h \). Let the set \( K_0 = [x_j]_{j=h}^{\infty} \), of values of the \( x_j \), once the periodicity has started, be a subset of \( J \), and let the number of distinct values in it be \( P = |K_0| = \lambda \). Then a sufficient condition for the sequence \([x_j]_{j=0}^{\infty} \) to be uniform in \( J \), is that there be integers \( \alpha \) and \( p \), with \( 0 \leq p \leq M \), such that
\[
\lambda = 2^{M-p} \quad \text{and} \quad (\forall j \geq h) \ 2^p \mid (x_j - \alpha). \tag{68}
\]
\<<\text{Since the sequence} [x_j]_{j=0}^{\infty} \text{is periodic, with period} \lambda, \text{starting at index} h; \text{the sequence} [x_j - \alpha]_{j=0}^{\infty} \text{, offset from the first by} -\alpha, \text{is also periodic, with the same period} \lambda, \text{starting at the same index} h, \text{as is noted after Definition 2. That the set} K_0 \text{has just} \lambda \text{distinct elements indicates that, in the period, there are no repeated values. Now, let} \>>>
$K_\alpha = \{<x_j - \alpha|\Omega>\}_{j=1}^\infty$ be the set of offset periodic values, reduced modulo $\Omega$; clearly, these are also just $\lambda$ in number. If we write

$$J_0 = J \quad \text{and} \quad J_p = \{r \cdot 2^p: 0 \leq r \leq 2^{M-p} - 1\},$$

then $J_p$ is obviously the set of all integer multiples of $2^p$ in $J$ (and so in $L$). Hence, the total number of such multiples is $2^{M-p}$, and $J_p$ is CES in $L$ (by Definition 3, since adjacent points are $2^p = (Q - 0)/2^{M-p}$ apart). If (68) holds, then $K_\alpha$ is clearly a subset of $J_p$, since $2^p$ divides every $x_j - \alpha$, and so, since $\lambda = 2^{M-p}$, $K_\alpha$ must equal $J_p$. Thus, $K_\alpha$ is CES in $L$; and therefore so is the original set $K_0$, offset from $K_\alpha$ by $+\alpha$, as is noted after Definition 3. Thus, by Definition 4, the sequence $\{x_j\}_{j=0}^\infty$ is uniform in $J$, with coarseness $\Omega/\lambda$.

**Lemma 17.** The period, $\lambda$, of the sequence $\{x_j\}_{j=0}^\infty$ generated by (4) equals the number, $P = |K_0|$, of distinct values in the periodic set $K_0 = \{x_j\}_{j=1}^\infty$.

(We refer to the proof of Lemma 7. The $j - i$ values $x_i, x_{i+1}, x_{i+2}, \ldots, x_{j-1}$ are all different, and thereafter the values repeat, because, by (2) or (4), equal predecessors in the sequence have equal immediate successors, and because $x_i = x_j$. Thus, $P = j - i$. Therefore, by Lemma 4, $P$ is a multiple of $\lambda$. But, since all $P$ values in the above list differ, $\lambda$ cannot be less than $P$; whence $\lambda = P$.)

We now have all the facts we need to prove our main result.

**Theorem 1.** If the set $J$ is defined by (28) and the sequence $\{x_j\}_{j=0}^\infty$ is generated by (4) with odd parameter $\alpha$ satisfying (50) and (52); then the sequence is uniform in $J$, in the sense of Definition 4. When $g$ is defined by (39) and (40), the coarseness of the sequence is given by

(i) $2^c$, if $c \leq s + d - 1$ and $c < \infty$;

(ii) $\min\{2^M, 2^g\}$, if $c = s + d$ and $c < \infty$;

(iii) $\min\{2^M, 2^{s+d}\}$, if $c \geq s + d + 1$ or $c = s = \infty$.
Lemma 9 tells us that the sequence \( [x_j]_{j=0}^{\infty} \) generated by (4) is completely periodic, since the parameter \( a \) is odd; and Lemma 17 tells us that the number, \( P \), of distinct values of \( x_j \) in the period of the sequence equals its period, \( \lambda \) (i.e., the period consists of \( \lambda \) different values, with no repetitions). Lemma 16 gives sufficient conditions for the sequence to be uniform; and, in the present case, all of that lemma's preliminaries are satisfied, with \( h = 0 \), \( K_0 = [x_j]_{j=0}^{\infty} \), and \( P = |K_0| = \lambda \). By (53) [which holds for all \( b \) (see Lemma 10)], \( \lambda \) takes the form \( 2^u \) with \( 0 \leq u \leq M \); which translates, if we write \( u = M - p \), into the first part, \( \lambda = 2^{M-p} \), of the condition (68) of Lemma 16. Further, Lemmas 12 and 14 specify the corresponding values of \( p \). Therefore, the second part of the condition (68), which becomes

\[
(\forall j \geq 0) \quad 2^p \mid (x_j - \alpha),
\]

alone remains to be verified, with the help of Lemma 15. Given that the sequence is indeed uniform in \( J \), it then follows from Definition 4 that the coarseness of the sequence is \( Q/P = Q/\lambda = 2^M/2^{M-p} = 2^p \).

An examination of Lemmas 12, 14, and 15 indicates that there are six cases to be considered. Necessary correspondences between cases are tabulated below. Cases (I), (II), and (III) correspond to Part (i) of our theorem; Case (IV), to Part (ii); and Cases (V) and (VI) to Part (iii).

| Table 1 |
|-----------------|----------------|----------------|----------------|
| **Case**       | **Lemma 12**   | **Lemma 14**   | **Lemma 15**   |
| \( c < \infty \): |                |                |                |
| (I) \( c \leq s - 1 \) | (i) \( p = c < M \) | \( - \)       | (a)         |
| (II) \( c = s \) | (i) \( p = c < M \) | \( - \)       | (b)         |
| (III) \( s + 1 \leq c \leq s + d - 1 \) | (i) \( p = c < M \) | \( - \)       | (c)         |
| (IV) \( c = s + d < g \) | (ii) \( p = \min\{M, g\} \) | \( - \)       | (c)         |
| \( c < \infty \) or \( c = \infty \): |                |                |                |
| (V) \( c \geq s + d + 1 \) | (iii) \( p = s + d < M \) | \( p = \min\{M, s + d\} \) | (c)         |
| (VI) \( c = s = \infty \) | \( - \)       | \( p = M \)    | (c)         |
(I) If \( c < \infty \) and \( c \leq s - 1 \), then we have Case (a) of Lemma 15: members \( x_{2j} \) of the sequence \( \{x_j\}_{j=0}^\infty \) are even multiples of \( 2^c \), and members \( x_{2j+1} \) are odd multiples of \( 2^c \); so all \( x_j \) are multiples of \( 2^c \). Thus, (70) holds if we take \( \alpha = 0 \) and apply \( p = c \) (from Lemma 12). The sequence is therefore uniform in \( J \), with coarseness \( 2^c \).

(II) If \( c = s < \infty \), then we have Case (b) of Lemma 15: members \( x_{2j} \) of the sequence are odd multiples of \( 2^c \), and members \( x_{2j+1} \) are even multiples of \( 2^c \); so that, again, all \( x_j \) are multiples of \( 2^c \); whence, as before, since \( p = c \), the sequence is uniform in \( J \), with coarseness \( 2^c \).

In all remaining cases, we have Case (c) of Lemma 15: all the \( x_j \) are odd multiples of \( 2^s \). Also, for all \( j \), by equation (7), \( \alpha^j - 1 = (\alpha - 1)S_j(a) \); so that, by Lemma 6, with equation (39),

\[
x_j = \langle x_0 + (\alpha^j - 1)x_0 + S_j(a) b \mid Q \rangle = \langle x_0 + S_j(a) W \mid Q \rangle;
\]
whence, for some \( C_j \), \( x_j - x_0 = S_j(a)W + C_j 2^M \); so that, by (40),

\[
(\forall j \geq 0) \quad 2^{\min\{g, M\}} \mid (x_j - x_0).
\]

(III) If \( c < \infty \) and \( s + 1 \leq c \leq s + d - 1 \), then, again, \( p = c \). By (39), (40), and (41), since \( c < s + d \), \( g = c < M \); whence \( c = \min\{g, M\} \). If we take \( \alpha = x_0 \), (70) follows from (72); so that the sequence is, once again, uniform in \( J \), with coarseness \( 2^c \). This completes the proof of Part (i) of our theorem.

(IV) If \( c = s + d < \infty \), then, by Lemma 12, \( p = \min\{M, g\} \), where \( g \) is defined by (39) and (40). Taking \( \alpha = x_0 \), we see that (70) holds, by (72); whence the sequence \( \{x_j\}_{j=0}^\infty \) is uniform in \( J \), with coarseness \( 2^p = \min\{2^M, 2^s\} \). This proves Part (ii) of our theorem.

(V) If \( c \geq s + d + 1 \), then either \( c < \infty \) and \( p = s + d \leq M - 2 \), by Lemma 12(iii); or \( c = \infty \) and \( p = \min\{M, s + d\} \), by Lemma 14. Thus, if \( p = M \leq s + d \), we have \( \lambda = 1, b = 0 \), and \( [\text{since we are not in Case (VI)}] \); i.e., since \( x_0 \neq 0 \) \( s < M \), by (41); and \( 1 \leq M - s \leq d \), which is possible, by (52). In this case, all the \( x_j \) are equal; whence we see that every \( x_j - x_0 = 0 \), which is divisible by \( 2^M = 2^p \). Thus, taking \( \alpha = x_0 \), we obtain (70); so that our sequence will be uniform in \( J \), with coarseness \( 2^p = 2^M = \min\{2^M, 2^{s+d}\} \).
If \( p = s + d < M \), on the other hand, then \( c \) may be finite or infinite. By Lemma 15(c), write

\[
x_j = 2^s X_j,
\]

where every \( X_j \) is an odd number. Then, by (2),

\[
X_{j+1} \equiv aX_j + 2^{-s} b \pmod{2^{M-s}},
\]

where, by (52), \( M - s > d \geq 2 \), and \( 2^{-s} b \) is divisible by \( 2^d \) [by (40), since, in the present case, \( c - s \geq d + 1 \)]. Hence, by (51) and (58),

\[
X_{j+1} \equiv X_j \pmod{2^d};
\]

so that all the \( X_j \) are not only odd, but congruent to the same odd number, modulo \( 2^d \). This means that every \( x_j \) equals \( x_0 = 2^s X_0 \) plus a multiple of \( 2^{s+d} = 2^p \). Taking \( \alpha = x_0 \), we see that (70) holds; and therefore, again, the sequence \( \{x_j\}_{j=0}^{\infty} \) is uniform in \( J \), with coarseness \( 2^p = 2^{s+d} = \min\{2^M, 2^{s+d}\} \).

(Vi) Finally, if \( c = s = \infty \), then \( b = x_0 = 0 \), whence, by (2), every \( x_j = 0 \); hence \( \lambda = 1 \) and so \( p = M \). By the same token, (70) holds for \( \alpha = 0 \); so that our sequence is indeed uniform in \( J \), with coarseness \( Q \). This completes the proof of our theorem. \\

**Corollary 1.** The coarseness of the sequence \( \{x_j\}_{j=0}^{\infty} \), defined as in Theorem 1, attains its minimum possible value, namely, 1, if and only if \( c = 0 \).

\( \ll \) It is clear from the definitions (39) and (40) underlying Theorem 1 that \( s \geq 0 \) and \( c \geq 0 \). By (50) and (52), and since \( a \in J \),

\[
5 \leq 2^d + 1 \leq (2r - 1) 2^d + 1 = a < 2^M; \text{ whence}
\]

\[
M \geq 3.
\]

In Case (i) of the theorem, the coarseness \( 2^c = 1 \) only when \( c = 0 \); implying that \( s + d - 1 \geq 0 \) and thus in no way restricting the allowable values of \( s \) [since \( s \geq 0 \) anyway, and, by (52), \( d \geq 2 \)]. In Case (ii), \( g \geq c + 1 = s + d + 1 \) and the coarseness is \( \min\{2^M, 2^d\} \). By (75), \( 2^M \geq 8 \); and, since \( s \geq 0 \), \( g \geq d + 1 \geq 3 \), by (52), so that \( 2^g \geq 8 \). Thus, either way, the coarseness is at least 8. In Case (iii), similarly, by (75), and because \( s \geq 0 \) and \( d \geq 2 \), the coarseness \( \min\{2^M, 2^{s+d}\} \) is at least 4. Thus, the absolutely best coarseness, 1, is attained when and only when \( c = 0 \) [in Case (ii)]. \( \gg \)
Corollary 2. Given the set \( F \) defined in (29) and the sequence \( \{x_j\}_{j=0}^{\infty} \), defined by (1) and (2), with parameter \( a \) satisfying (50) and (52); the sequence is uniform in \( F \), in the sense of Definition 4, and the coarseness of the sequence is given by the values in Cases (i), (ii), and (iii) of Theorem 1.

Both sets, \( J \) and \( F' \), have \( Q \) members (points) and are respectively CES, in \([0, Q]\) and \([0, 1]\). The sequence \( \{x_j\}_{j=0}^{\infty} \) stands in the same relation to \( J \) as does \( \{\xi_j\}_{j=0}^{\infty} \) to \( F' \), and the corresponding sets \( K_0 = \{x_j\}_{j=0}^{\infty} \) and \( K_1 = \{\xi_j\}_{j=0}^{\infty} \) both have just \( \lambda \) members. Thus, by Definition 4 and Theorem 1, the corollary follows.

We have now collected sufficient information, on the uniformity properties of linear-congruential pseudo-random sequences, to enable us to move on to the main purpose of our study; namely, the generation and analysis of tree-structured families of generators. We shall discover that the results, embodied, for the most part, in Theorem 1 and its corollaries, which tell us about the uniformity and coarseness of a single sequence, suffice to analyze the properties of independence and consonance between members of families of such sequences.

4. TREE-STRUCTURED FAMILIES OF GENERATORS

We now proceed to consider tree-like branching processes. We take particle-transport problems as important and typical paradigms. The model often used has two kinds of random steps: those representing the rectilinear (or, in the presence of force-fields, curved) particle flight across the empty space of which all materials are overwhelmingly composed (a statistical Poisson distribution of path-length, determined by the 'mean free path', is used to sample the distance traveled); alternating with steps representing 'collision' events, terminating such free flights. Collision events include elastic or inelastic rebound-collisions and various nuclear reactions, which often generate new particles (of matter or radiation); these last lead to a branching of the particle histories. The creation of 'virtual particles' (used, for example, in the Monte Carlo 'particle-splitting' technique, and in obtaining Monte Carlo scores at small-aperture detectors) also leads to branching. (Of course, in most Monte Carlo computations, all
'particles' are more-or-less virtual!) Since each step in a particle history (or random walk) may typically require about 10 random numbers, we may expect our pseudo-random sequence to entail branching at every \( T \)-th term, where \( T \) is of the order of 10. While it is certainly feasible to allow branching at every random number, it is likely to be more economical to pick such a \( T \) and only allow branching at every \( T \)-th step of the random sequence. The price we pay is that \( T \) must be an over-estimate, so as to ensure that, at least, most of the time, \( T \) random numbers suffice to compute a random-walk step [if more are needed, in a particular step, then we must allocate an integer multiple of \( T \) random numbers to this step]; thus, quite a few random numbers will be wasted in the process.

Before we can move forward, we must consider the behavior of the sequence \( \{x_{T-1}, x_{2T-1}, x_{3T-1}, x_{4T-1}, \ldots \} = \{x_{JT-1}\}_{j=1}^{\infty} \) corresponding to the branch-points of the process (\( x_{JT-1} \) is the current pseudo-random number last obtained, when \( T \) numbers have been generated and a branch may occur).

**Lemma 18.** The behavior of the sequence \( \{x_{JT-1}\}_{j=1}^{\infty} \) of branch-points is given by

\[
X_{j+1} = \langle A X_j + B | Q \rangle,
\]

when we write

\[
A = \langle a_T | Q \rangle, \quad B = \langle S_T(a) b | Q \rangle, \quad \text{and} \quad X_j = x_{JT-1}.
\]

\([77]\)

\(<\langle \text{By Lemma 6, the relation (30) holds; so that, using (5), we see that, modulo } Q, \rangle\)

\[
x_{(j+1)T-1} = a_{(j+1)T-1} x_0 + S_{(j+1)T-1}(a) b
\]

\[
= a_{(j+1)T-1} x_0 + (a_{(j+1)T-2} + a_{(j+1)T-3} + \ldots + a^2 + a + 1) b
\]

\[
= a^T \left[ a^{T-1} x_0 + (a^{T-2} + a^{T-3} + \ldots + a^2 + a + 1) b \right]
\]

\[
+ (a^{T-1} + a^{T-2} + a^{T-3} + \ldots + a^2 + a + 1) b
\]

\[
= a^T \left[ a^{T-1} x_0 + S_{T-1}(a) b \right] + S_T(a) b
\]

\[
= a^T x_{JT-1} + S_T(a) b.
\]

[78]

With the notations of (3) and (77), (78) takes the form (76).>>
The recurrence relation (76) is exactly of the same form as (4); so that all our earlier analysis applies here, and Theorem 1 applies to the sequence \([X_j]_{j=0}^\infty\) just as it does to \([x_j]_{j=0}^\infty\). By Corollary 1, we observe that odd values of \(B\) are preferable; and, clearly, by (77) and Lemma 3 [with \(q = 1\), by (48)], \(B\) will be odd, if and only if both \(b\) and \(T\) are odd. It is easily seen, by (51), that

\[
A = a^T = 1 \pmod{2d},
\]

for all values of \(T\). Henceforth, we shall revert, throughout, to the more familiar notation of (4), rather than that of (76); but with the understanding that an equally-spaced subsequence \([x_{jT-1}]_{j=1}^\infty\) of \([x_j]_{j=0}^\infty\) may well be what we are really dealing with.

The recurrence relation (4), with parameters \(a\), \(b\), and \(x_0\) (we take \(Q\) and \(M\) as fixed), generates a linear-congruential sequence \([x_j]_{j=0}^\infty\) of integers in \(J\). It constitutes a pseudo-random generator, which we may denote by \(\Phi = \mathcal{S}(a, b, x_0)\). Having analyzed the periodic behavior and uniformity of a single linear congruential sequence, we can now consider a pair of such sequences: (i) \([x_j]_{j=0}^\infty\), with generator \(\Phi = \mathcal{S}(a, b, x_0)\), characterized by (4), and (ii) \([x^\dagger_j]_{j=0}^\infty\), with generator \(\Phi^\dagger = \mathcal{S}(a^\dagger, b^\dagger, x^\dagger_0)\), say, characterized by

\[
(\forall j \geq 0) \quad x^\dagger_{j+1} = \langle a^\dagger x^\dagger_j + b^\dagger | Q \rangle.
\]

We may now define the difference-sequence \([\delta_{j\dagger}]_{j=0}^\infty\) as we did in (26), and observe at once that

\[
(\forall j \geq 0) \quad \delta_{j+1} = \langle a\delta_j + (a - a^\dagger)x^\dagger_j + (b - b^\dagger) | Q \rangle.
\]

By applying (71) to both \([x_j]_{j=0}^\infty\) and \([x^\dagger_j]_{j=0}^\infty\) in (26), we get that

\[
\delta_n = \langle \delta_0 + S_n(a) W - S_n(a^\dagger) W^\dagger | Q \rangle,
\]

where \(W^\dagger = (a^\dagger - 1)x^\dagger_0 + b^\dagger\) is the counterpart, for the generator \(\Phi^\dagger\), of \(W\), defined in (39). This formula is rather difficult to analyze for the period and uniformity of the difference-sequence; but a particular case
proves to be more tractable. Suppose that we restrict our consideration to \( \alpha^t = \alpha \); then (81) becomes

\[
(\forall j \geq 0) \quad \delta_{j+1} = \langle a \delta_j + (b - b^t) | \mathcal{G} \rangle.
\]

which is exactly similar to (4), except that \( b \) is replaced by \( \beta = \langle b - b^t | \mathcal{G} \rangle \). It follows that all the results obtained so far (up to and including Theorem 1 and its corollaries) for the sequence \([x_j]_{j=0}^{\infty}\) also apply to the sequence \([\delta_j]_{j=0}^{\infty}\). It is just another linear-congruential sequence, whose generator may be written as \( \Lambda = \mathcal{S}(a, \beta, \delta_0) \).

All this can now be generalized to a family of generators, which we may denote by \( \Phi_\mu = \mathcal{S}(a_\mu, b_\mu, x_\mu 0) \), with parameters \( a_\mu, b_\mu, \) and \( x_\mu 0 \), satisfying

\[
(\forall j \geq 0) \quad x_{\mu(j+1)} = \langle a_\mu x_{\mu j} + b_\mu | \mathcal{G} \rangle.
\]

We restrict our consideration, by taking \((\forall \mu) a_\mu = a\), and write

\[
\beta_{\mu \nu} = \langle b_\mu - b_\nu | \mathcal{G} \rangle \quad \text{and} \quad \delta_{\mu \nu j} = \langle x_{\mu j} - x_{\nu j} | \mathcal{G} \rangle.
\]

Then

\[
(\forall j \geq 0) \quad \delta_{\mu \nu(j+1)} = \langle a \delta_{\mu \nu j} + \beta_{\mu \nu} | \mathcal{G} \rangle.
\]

It is reasonable to minimize the coarseness of each individual sequence; and, by Corollary 1, the absolute minimum, 1, is attainable when and only when every \( c_\mu = 0 \), i.e., every \( b_\mu \) is odd. The values of the parameters \( x_{\mu 0} \) and \( a \), subject only to (50) and (52), are arbitrary. This means that we have at our disposal fully half of all possible linear-congruential sequences (altogether \( 2^{M-1} \) sequences) for each choice of \( x_0 \), when \( a \) is fixed. However, this does entail that every \( \beta_{\mu \nu} \) will now be even. (There is no choice of more than two integers \( b_\mu \) which will permit us to get all odd \( \beta_{\mu \nu} \).

Now let us consider the kind of branching random walk for which the present study is intended to provide effective pseudorandom generators. In Figure 1, we see the first five levels of a binary tree with the nodes numbered in a simple, systematic manner. The caption explains the system. From any odd-numbered node, say \( N_\mu = 2\mu + 1 \), \((\mu = 0, 1, 2, \ldots)\), we define a random walk, or sequence of nodes,
\[
\Gamma_\mu = [N_\mu \rightarrow 2N_\mu \rightarrow 4N_\mu \rightarrow \ldots \rightarrow 2^mN_\mu \rightarrow \ldots].
\] (87)

obtained by taking the left-slanting branch at every node (i.e., going from parent to left-child, every time), which will correspond, for example, in the case of a particle-transport problem, to a single particle-track.

![Binary Tree Structure](image)

**Figure 1.**

**Binary Tree Structure.**

Level \(k\) has \(2^k\) nodes, numbered (boldface, next to node-circle) \(2^k, 2^k + 1, 2^k + 2, \ldots, 2^{k+1} - 1\). Children of node number \(n\) are nodes numbers \(2n\) (on left) and \(2n + 1\) (on right). *Left branches* are shown thicker; they denote continuing random walks \(\Gamma_\mu\) [index shown in node-circles; \(\mu\) in (88)], generated by single linear-congruential generators.

Associated with the walk \(\Gamma_\mu\), there will be, at each node, an array or other data-structure, giving the properties of the corresponding event, e.g., of a collision in the particle-history. The statistical samples occurring at every node of the random walk will be computed using pseudo-random numbers coming from a single generator of type \(\Phi_\mu = \mathcal{G}(a, b_\mu, x_{\mu0})\), satisfying (84), with parameters \(x_{\mu0}, b_\mu,\) and \(a_\mu = a\), satisfying (50) and (52). When an additional particle is generated at node number \(v\), this will correspond to taking a right-slanting branch, to the child-node numbered \(N_v = 2v + 1\), where a new pseudo-random generator \(\Phi_v = \mathcal{G}(a, b_v, x_{v0})\), with parameters \(a, b_v,\) and \(x_{v0}\), initiates a new, concurrent particle-track or random walk, \(\Gamma_v\).
Since it is typical that branching does not actually occur at every node (and, indeed, since, as has been explained in §1, it would be totally impossible, in practice, to perform the computations needed if every branch did occur), it is of great practical utility, that the generator \( \Phi_v \), needed on branching at node number \( v \), should be identified by appeal only to the index \( v \), or, at worst, to a small number of parameters computed and stored at the node \( v \).

Let \( v \) be a node in \( \Gamma_\mu \); so that, by (87), for some integer \( m \),
\[
\nu = 2^m(2\mu + 1) \quad \text{and} \quad 2^m \uparrow v.
\] (88)

Then we may associate, with the node \( v \), a record,
\[
R_v = (2^m N \mu, b_\mu, x_{\mu m}).
\] (89)

consisting of (i) the current node number, \( \nu = 2^m N \mu \) (from which both \( m \) and \( \mu \) can be uniquely determined); (ii) the value of the parameter \( b_\mu \) of the current generator \( \Phi_\mu \) (remember that the parameter \( a \) is supposed to be common to all random generators in this scheme, or family); and (iii) the current random number \( x_{\mu m} \). We now begin the new random walk \( \Gamma_v \), with new parameters, \( b_v \) and \( x_{v0} \), and the particular scheme that we adopt is specified when we define the functional relationships between these new parameters and the record:
\[
b_v = B(2^m N \mu, b_\mu, x_{\mu m}) = B(R_v),
\] \[
x_{v0} = X(2^m N \mu, b_\mu, x_{\mu m}) = X(R_v).
\] (90)

This can also be formalized by putting:
\[
R_{Nv} = (N_v, b_v, x_{v0}) = M(2^m N \mu, b_\mu, x_{\mu m}) = M(R_v).
\] (91)

The mapping \( M \) (or, more explicitly, the functions \( B \) and \( X \) comprising it) determine the particular algorithm we choose.

Consider, first, the relationship between two segments of the same random sequence \( [x_j]_{j=0}^\infty \), say one beginning at \( x_0 \) and the other at \( x_H \). Then we may take
\[
x^j = x_{H+j} \quad \text{and} \quad \Phi^j = S(a, b, x_H);
\] (92)
so that, by (83) with \( b^t = b \), we see that the sequence \( [\delta_j]_{j=0}^{\infty} \) defined in (26) has generator \( \Delta = S(a, 0, \sigma_0 - \sigma_H) \).

**Lemma 19.** Given the set \( J \) defined in (28), the sequence \( [x_j]_{j=0}^{\infty} \) generated by (4) with parameter \( a \) satisfying (50) and (52), and given any positive integer \( H \); the sequences \( [x_j]_{j=0}^{\infty} \) and \( [x_j]_{j=H}^{\infty} \) (which differ only by the positional offset \( H \)) are independent with respect to \( J \), in the sense of Definition 5. When \( c, s, d, \) and \( g \) are defined by (39) and (40), and \( \kappa \) by

\[
2^\kappa \uparrow H, \tag{93}
\]

the two sequences have consonance \( \min(2^M, 2^{\kappa+g+d}) \).

By applying Theorem 1 to the generator \( \Delta \), we see at once that the sequence \( [\delta_j]_{j=0}^{\infty} \) is uniform in \( J \); and therefore, by Definition 5, we immediately conclude that the two sequences \( [x_j]_{j=0}^{\infty} \) and \( [x_j]_{j=H}^{\infty} \) are independent with respect to \( J \). Since, by (13), \( 2^{\infty} \uparrow 0 \), and the second parameter of the generator is 0, the corresponding 'power of divisibility' of that parameter is \( \infty \); so that, by Theorem 1, the coarseness of \( [\delta_j]_{j=0}^{\infty} \) is \( G = \min(2^M, 2^{\kappa+g+d}) \), where \( d \) is defined by (50) and (52), and \( \sigma \) is defined by \( 2^\sigma \uparrow (x_0 - x_H) \). Hence, by Definition 5, the consonance of \( [x_j]_{j=0}^{\infty} \) and \( [x_j]_{j=H}^{\infty} \) is \( G \).

Now, by (15) with (48), \( 2^\kappa \uparrow S_H(a) \); by (39) and (40), \( 2^g \uparrow W \); and, finally, by (71), \( <x_0 - x_H> = <S_H(a)W|G> \). Therefore, we see that \( \sigma = \kappa + g \); and so \( G = \min(2^M, 2^{\kappa+g+d}) \).

Just as we stipulated, first, that the parameter \( a \) be odd, and then that it should satisfy (50) and (52), so as to minimize the coarseness [that is, maximize the uniformity] of the individual sequences; so we now seek to minimize the consonance \( G \) of a pair of sequences. To this end, we may minimize \( d \), subject to (52), by

\[
d = 2. \tag{94}
\]
so that, by (50), this is equivalent to \( a = (2r' - 1)4 + 1 = 8(r' - 1) + 5 \), or

\[
a = 5 \pmod{8}. \tag{95}
\]

**Corollary 3.** Under the conditions of Lemma 19, if we impose the additional constraint (95), and choose the parameter \( b \) to be odd; then the consonance of the two sequences becomes \( \min(2^M, 2^{x+2}) \).

<<Since (95) is equivalent to (94), direct substitution of 2 for \( d \) in the formula given by the lemma yields \( G = \min(2^M, 2^{x+g+2}) \). Since \( b \) is made odd, so that \( c = 0 \), we have \( c < s + 2 = s + d, \) by (94); whence Case (i) of Lemma 11 yields that \( g = c = 0 \). The corollary now follows immediately.>>

Warnock (see WAR 83) proposes, in our notation, that all 'left-slanting' generators \( \Phi_\mu \) should share common parameters \( a \) and \( b \). Thus, his function of type \( B \), say \( B_{\Phi\Psi} \), is the projection of the second argument, unchanged:

\[
b_{\nu} = B_{\Phi\Psi}(2^m N_, b_\mu, x_\mu m) = b_\mu = b. \tag{96}
\]

His function of type \( X \), say \( X_{\Phi\Psi} \), applies a step of type (4), with its own independent parameters, \( a^t \) and \( b^t \), say, to go from the last random number \( x_\mu m \) to the first one of the new sequence:

\[
x_{\nu 0} = X_{\Phi\Psi}(2^m N_, b_\mu, x_\mu m) = \langle a^t x_\mu m + b^t | Q \rangle. \tag{97}
\]

Since all the left-slanting generators in Warnock's scheme have the same parameters \( a \) and \( b \), if we select \( a \) satisfying (95) [and so (50) and (52)] and \( b \) odd \( [c = 0] \); then, by Corollary 1, all the resulting sequences will be uniform in \( J \), with minimal coarseness 1. Thus, the period has length \( Q \); that is, every value in \( J \) occurs in each such sequence. Consequently, all the possible sequences are just positional offsets of each other; and therefore, by Corollary 3, if a pair of such sequences has positional offset \( H \) satisfying (93), it will exhibit the consonance \( \min(2^M, 2^{x+2}) \).

Unfortunately, it is impossible to improve the situation optimally by making all \( \kappa = 0 \). This is because of the structure of the integers, with respect to divisibility by powers of 2. The sequence of \( \kappa \)-values takes the form shown in Figure 2.
Figure 2.

Divisibility of the Integer Sequence by Powers of 2.

For a segment of the sequence of integers $H$, the corresponding values of $\kappa$, such that $2^\kappa > H$, are tabulated, with each row having a single value of $\kappa$. (The structure is reminiscent of the "Sieve of Eratosthenes" used to find prime numbers.) Observe that, to be as far as possible from $\kappa = \kappa_0$, one has to be close to a value of $H$ with $\kappa = \kappa_0 - 1$. For example, with $x_0 = 4$ and $H$ lying between 16 and 32, it is best to choose $H = 21$ or 23, somewhat closer to the lesser of the extreme $\kappa$-values, 4 and 5. The two extreme values will, of course, never be equal.

We can, at best, hope that sequences corresponding to immediately adjacent events will have low values of $\kappa$. The generator $\Phi_\mu$ begins at the node numbered $N_\mu = 2\mu + 1$ and passes, in a left-slaunting direction, through the node numbered $\nu = 2^m N_\mu$, from which a branch goes to its right-child node, numbered $N_\nu = 2\nu + 1$. The generator $\Phi_\nu$ begins there, and its sequence $[x_{\nu l}]^\infty_{l=0}$, in Warnock's scheme, is, as we have seen, just a positionally-offset copy of the sequence $[x_{\mu l}]^\infty_{l=0}$ of generator $\Phi_\mu$. We certainly want the two histories, beginning at the left and right children of node $\nu$, to have the least possible consonance; so we would like the offset, between $x_{\mu (m+1)} = \langle a x_{\mu m} + b | \mathcal{Q} \rangle$ and $x_{\nu 0} = \langle a^\dagger x_{\mu m} + b^\dagger | \mathcal{Q} \rangle$, to be odd; this is clearly equivalent to having the offset between $x_{\mu m}$ and $x_{\nu 0}$ even. By an obvious extension of (30), we require that there be an integer $n$, such that

$$a^\dagger x_{\mu m} + b^\dagger = a^{2n} x_{\mu m} + S_{2n}(a) b \pmod{\mathcal{Q}}.$$  

(98)
or, by (3), \( a^t = \langle a^{2n} | Q \rangle \) and \( b^t = \langle S_{2n}(a)b | Q \rangle \). \( (99) \)

Since \( (99) \) is independent of \( x_{\mu n} \), we see that a single transformation of the form \( (97) \) will work in all cases.

However, our advantage is somewhat brittle. It is physically desirable that the track, generated by \( \Phi_\nu \) and beginning at node \( N_\nu \), should also have small consonance with tracks beginning at nodes neighboring node \( 2\nu \) in the chain generated by \( \Phi_\mu \); i.e., corresponding to sequential (positional) offsets close to, but different from, \( 2n - 1 \) (with \( n \) the same as that in \( (99) \)). These offsets will be even, in about half the cases, and examination of the sequence of \( \kappa \)-values in Figure 2 indicates that, if we wish to avoid \( \kappa \geq \kappa_0 \), say, we shall certainly have a near-neighbor with \( \kappa = \kappa_0 - 1 \). As for more distant tracks, across the tree, these will have a variety of offsets, but this is hardly to be avoided. After all, we are looking at a universe of only \( 2^M \) distinct sequences, to fill \( 2^{k-1} \) tracks [left-slanting branches], in a binary tree of height \( k \), with \( 2^k - 1 \) branch-points and \( 2^{k+1} \) nodes. Since a typical value of this \( k \) is perhaps \( 10^2 - 10^6 \), while a typical value of \( M \) is about 48, the capacity of the scheme is evidently overloaded.

The plausible argument, that computational runs requiring some \( 10^3 - 10^7 \) random numbers should be pretty unrelated, when taken from random segments of a pseudo-random sequence with period of the order of \( 2^{48} = 3 \times 10^{14} \), at least thirty million times longer, turns out not to be entirely valid. However, in mitigation, it should be pointed out that, until now, no rigorous analysis of the algorithm was available.

If one nevertheless decides to adopt this scheme, the indication is strong that one should adopt \( a \) satisfying \( (95) \), \( b \) odd, and \( a^t \) and \( b^t \) satisfying \( (99) \), with values of \( n \) such as 11, 12, 22, 23, or 24 (for \( H = 21, 23, 43, 45, \) or 47, respectively).

We now leave Warnock's algorithm, and return to our consideration of the more general relationship between two sequences, \( [x_j]_{j=0}^\infty \) and \( [x_j^t]_{j=0}^\infty \), whose respective generators are

\[ \Phi = \mathcal{S}(a, b, x_0) \] and \( \Phi^t = \mathcal{S}(a, b^t, x_0^t) \), and whose difference \( [\delta_j]_{j=0}^\infty \), with \( \delta_j = \langle x_j - x_j^t | Q \rangle \), satisfies \( (83) \). With \( \beta = \langle b - b^t | Q \rangle \neq 0 \). We shall assume that both \( b \) and \( b^t \) are odd integers, and that \( a \) satisfies \( (95) \). Following \( (39) \) and \( (40) \), let
\[ \Omega = \langle \delta_1 - \delta_0 \mid Q \rangle = \langle (a - 1)\delta_0 + \beta \mid Q \rangle. \]  

(100)

and 

\[ 2^\gamma \uparrow \beta, \ 2^\sigma \uparrow \delta_0, \ \text{and} \ 2^\tau \uparrow \Omega. \]  

(101)

By Theorem 1 and Definition 5, we now obtain:

**Theorem 2.** Given the set \( J \) defined in (28) and the parameters \( a \), with (95), and \( b \) and \( b^t \), both odd, with \( \langle b - b^t \mid Q \rangle \neq 0 \); the sequences \( [x_j]_{j=0}^{\infty} \) and \( [x_j']_{j=0}^{\infty} \), with generators \( \Phi = \Phi(a, b, x_0) \) and \( \Phi^t = \Phi(a, b^t, x_0^t) \), respectively, are independent with respect to \( J \). If \( \Omega, \gamma, \sigma, \) and \( \tau \) are defined as in (100) and (101), then the consonance of the sequences is given by

(i) \[ 2^\gamma, \]  
if \( \gamma \leq \sigma + 1 \);

(ii) \[ \min(2^M, 2^\tau) \]  
with \( \tau > \gamma, \)  
if \( \gamma = \sigma + 2 \);

(iii) \[ 2^{\sigma+2}, \]  
if \( \gamma \geq \sigma + 3 \).


By (83) and (95), which implies (94), we have the conditions of Theorem 1, with \( d = 2 \), and \( \Omega, \gamma, \sigma, \) and \( \tau \) respectively taking the places of \( W, c, s, \) and \( g \). By our assumptions,

\[ 1 \leq \gamma < M; \]  

(102)

whence the case of \( \gamma = \sigma = \infty \) is impossible, and \( M > \gamma > \sigma + 2 \) [compare Lemma 12]. Theorem 2 follows immediately.

**Corollary 4.** Under the conditions of Theorem 2, if \( \delta_0 = x_0 - x_0^t = 0 \), then the consonance of the sequences is \( 2^\gamma \).

\[ \langle \delta_0 = 0, \text{then, by (101) with (13), } \sigma = \infty. \text{ Thus, by (102), we are in Case (i) of Theorem 2; and the corollary is immediate.} \]

It is instructive to note the dependence on \( \sigma \), for any given \( \gamma \), of the consonance determined by Theorem 2. This is sketched in Figure 3.
**Figure 3.**

Consonance as a Function of $\sigma$, for Fixed $\gamma$

The logarithm to base 2 of the consonance of two sequences, $[x_j]_{j=0}^\infty$ and $[x'_j]_{j=0}^\infty$, with generators $\Phi = \mathcal{R}(a, b, x_0)$ and $\Phi^\dagger = \mathcal{R}(a, b^\dagger, x_0^\dagger)$, respectively, is plotted against $\sigma$ (where $2^\sigma \uparrow \delta_0 = <x_0 - x_0^\dagger|\mathcal{G}>$, for given $\gamma$ (where $2^\gamma \uparrow \beta = <b - b^\dagger|\mathcal{G}> \neq 0$). Cases indicated are those used for classification of results in Theorem 2.

In the general situation described by (84) – (91), in which a family of generators $\Phi_\mu$, with a single common parameter $a$, satisfying (95), and with all their individual parameters $b_\mu$ odd, is matched to the odd-numbered nodes $N_\mu$ and left-slanting random walks $\Gamma_\mu$ of a binary tree; we seek, as ever, to minimize the consonance between the
sequences generated by the different $\Phi_{\mu}$' and especially between those sequences close to each other in the tree. The closest physical relationship will be between the sequences originating at the two child-nodes of any given node; for example, if the parent node is numbered $v$, the sequences are $[x_{\mu}]_{j=m+1}^{\infty}=[x_{\mu(i+m+1)}]_{i=0}^{\infty}$, beginning at node $2v$, and $[x_{v}]_{j=0}^{\infty}$, beginning at node $2v+1 = N_{v}$. If, in Theorem 2, we take

$$x_j = x_{\mu(m+j+1)}, \quad b = b_{\mu}, \quad x_j^\dagger = x_{v_j}, \quad b^\dagger = b_v; \quad (103)$$

so that

$$\beta = <b - b^\dagger | O> = <b_{\mu} - b_v | O> = \beta_{\mu v} \neq 0 \quad (104)$$

and

$$\delta_j = <x_j - x_j^\dagger | O> = <x_{\mu(m+j+1)} - x_{v_j} | O>; \quad (105)$$

then the conditions of the theorem are satisfied and the conclusions of the theorem hold, for all indices $v$ and functions $B$ and $X$.

Let us write

$$\Delta_{\mu v k} = <x_{\mu(m+h)} - x_{v0} | O> \quad (106)$$

(the notation makes sense, since, by (88), $\mu$ and $v$ determine $m$). Then we note, by (105), that, in particular,

$$\delta_0 = \Delta_{\mu v 1}. \quad (107)$$

We shall denote the logarithmic consonance (i.e., the logarithm to base 2 of the consonance) of our two sequences by $p_{\mu v 1}$.

**Corollary 5.** If $\Delta_{\mu v 1}$ is odd, then the logarithmic consonance of the sequences $[x_{\mu(i+m+1)}]_{i=0}^{\infty}$ and $[x_{v}]_{j=0}^{\infty}$ is given by

(i) $p_{\mu v 1} = 1$, \quad if $\gamma = 1$;

(ii) $2 < p_{\mu v 1} \leq M$, \quad if $\gamma = 2$;

(iii) $p_{\mu v 1} = 2$, \quad if $\gamma \geq 3$;

where $\gamma$ is defined by (101).
If $\Delta_{\mu \nu 1}$ is odd, then $\sigma = 0$, and, by (102), the three cases of Theorem 2 become those listed above; whence the values of $p_{\mu \nu 1}$ are as stated.

This result suggests that we should take $\Delta_{\mu \nu 1}$ odd and

$$\gamma \geq 3;$$

the latter condition is easily satisfied, e.g., by taking every $b_\mu = 1 \pmod{8}$. Note that, if $\Delta_{\mu \nu 1}$ is even and not zero, it is much harder to confine the values of $p_{\mu \nu 1}$.

Now consider, as we did for Warnock's scheme, what happens if we compare the sequences $[x_{\mu(t+m+H)}]_t$ and $[x_{\nu}]_t$, with a positional offset in one sequence. Then $\beta$ (and therefore also $\gamma$) is unaffected; but $\delta_0$ (and therefore also $\Omega$, $\sigma$, and $\tau$) will depend on $H$, since now

$$\delta_0 = <x_{\mu(m+H)} - x_{\nu 0} | Q> = \Delta_{\mu \nu H}.$$ (109)

Theorem 2 will clearly still apply. For different values of $H$, the logarithmic consonance of our two sequences, which is denoted by $p_{\mu \nu H}$, will depend on $\sigma$ as shown in Figure 3, with an isolated maximum-value 'spike' when $\sigma = \gamma - 2$ and $\gamma < p_{\mu \nu H} < M$. The dependence of $\sigma$ on $H$ will be scattered, rather as in Figure 2; and, as for Warnock's algorithm, this creates a problem.

By (71) and (106), we see that

$$\Delta_{\mu \nu h} = <\Delta_{\mu \nu 0} + S_h(a)W_{\mu \nu m} | Q>$$; (110)

where

$$W_{\mu \nu m} = <(a - 1)x_{\mu m} + b_\mu | Q>$$ (111)

is analogous to $W$ in (39). Thus, by (110) with $h = 1$,

$$\Delta_{\mu \nu 1} = <\Delta_{\mu \nu 0} + S_1(a)W_{\mu \nu m} | Q>.$$ (112)

Since, by (111) with (49) and because all the $b_\mu$ are going to be odd in our present discussion, $W_{\mu \nu m}$ is odd; and since, also, $S_1(a) = 1$; we see that

$$\Delta_{\mu \nu 1} \text{ is odd } \iff \Delta_{\mu \nu 0} \text{ is even.}$$ (113)
Further, by (93), (48), and Lemma 3, we have that $2^\kappa \uparrow S_H(a)W_{\mu m}$. Now, let us write

$$2^{\sigma_{\mu \nu H}} \uparrow \Delta_{\mu \nu H}. \quad (114)$$

Then, our usual line of argument [see, e.g., Lemmas 11 and 12], applied to (110) with $h = H$, yields that:

\[
\begin{align*}
(a) \quad \sigma_{\mu \nu H} &= \sigma_{\mu \nu 0} \quad \text{if } \sigma_{\mu \nu 0} < \kappa; \\
(b) \quad \sigma_{\mu \nu H} &> \kappa \quad \text{if } \sigma_{\mu \nu 0} = \kappa; \\
(c) \quad \sigma_{\mu \nu H} &= \kappa \quad \text{if } \sigma_{\mu \nu 0} > \kappa.
\end{align*}
\] (115)

Using Figure 4 as a guide, it is not too hard to derive, from (115) and Theorem 2 with $\sigma = \sigma_{\mu \nu H}$, the relationships shown in Table 2. Here, $D$ denotes the diameter of the cube bounded by coordinates $0$ and $M - 1$, in which the triangles $T_1$, $T_2$, and $T_3$ meet.

**Table 2**

<table>
<thead>
<tr>
<th>Region</th>
<th>Case in (115)</th>
<th>Case in Theorem 2</th>
<th>$p_{\mu \nu H}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>'1'</td>
<td>(a)</td>
<td>(iii)</td>
<td>$\sigma_{\mu \nu 0} + 2$</td>
</tr>
<tr>
<td>'2'</td>
<td>(c)</td>
<td>(iii)</td>
<td>$\kappa + 2$</td>
</tr>
<tr>
<td>'3'</td>
<td>(a), (b), (c)</td>
<td>(i)</td>
<td>$\gamma$</td>
</tr>
<tr>
<td>$T_1$</td>
<td>(c)</td>
<td>(ii)</td>
<td>$p &gt; \gamma$</td>
</tr>
<tr>
<td>$T_2$</td>
<td>(a)</td>
<td>(ii)</td>
<td>$p &gt; \gamma$</td>
</tr>
<tr>
<td>$T_3$</td>
<td>(b)</td>
<td></td>
<td>indeterminate</td>
</tr>
<tr>
<td>$D$</td>
<td>(b)</td>
<td>(i)</td>
<td>$\gamma$</td>
</tr>
</tbody>
</table>

**Theorem 3.** Define $\gamma$ by (101), $\kappa$ by (93), $\Delta_{\mu \nu H}$ by (106), and $\sigma_{\mu \nu H}$ by (114); and let $\Delta_{\mu \nu 1}$ be odd (i.e., $\sigma_{\mu \nu 1} = 0$). Then $\Delta_{\mu \nu 0}$ is even; and, if a clear minimum occurs (i.e., one of $\sigma_{\mu \nu 0} + 2$, $\kappa + 2$, and $\gamma$, is strictly smaller than the other two), then

$$p_{\mu \nu H} = \min[\sigma_{\mu \nu 0} + 2, \kappa + 2, \gamma]. \quad (116)$$
Logarithmic Consonance as a Function of $\sigma_{\mu \nu 0}$, $\kappa$, and $\gamma$.

The numbered solid regions, '1', '2', and '3', are pyramidal portions of the cube, bounded by faces of the cube and by triangular plane regions shaded otherwise than their own shading-key, and lying opposite to the similarly-shaded triangles: Region '1' is bounded by $T_2$ and $T_3$, and lies opposite to $T_1$; Region '2' is bounded by $T_3$ and $T_1$, and lies opposite to $T_2$; and Region '3' is bounded by $T_1$ and $T_2$, and lies opposite to $T_3$. The resulting values of the logarithmic consonance $p_{\mu \nu \kappa}$ are given in Table 2.
The first conclusion of the theorem, that $\Delta_{\mu\nu}0$ is even, follows at once from (113), since $\Delta_{\mu\nu}1$ is postulated to be odd. The result (116) follows immediately from Table 2 and the information in Figure 4, where we see that a "clear minimum" occurs precisely when we are in the interior of one of the regions '1', '2', or '3'.

Since, by Theorem 3 and (114), $\sigma_{\mu\nu}0 > 0$: when $H = 1$ (so that $\kappa = 0$), $\sigma_{\mu\nu}0 > \kappa$; whence $p_{\mu\nu}1 = \gamma$ if $\gamma < \kappa + 2 = 2$, and $p_{\mu\nu}1 = \kappa + 2 = 2$ if $\gamma > \kappa + 2 = 2$. Thus we recover Cases (i) and (iii) of Corollary 5.

**TABLE 3**

<table>
<thead>
<tr>
<th>$\sigma_{\mu\nu}0 + 2$ and $\gamma$</th>
<th>$\kappa$</th>
<th>Region</th>
<th>$p_{\mu\nu}H$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_{\mu\nu}0 + 2 &lt; \gamma$</td>
<td>$\kappa + 2 &lt; \sigma_{\mu\nu}0 + 2$</td>
<td>'2'</td>
<td>$\kappa + 2$</td>
</tr>
<tr>
<td></td>
<td>$\kappa + 2 = \sigma_{\mu\nu}0 + 2$</td>
<td>$T_3$</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>$\sigma_{\mu\nu}0 + 2 &lt; \kappa + 2 &lt; \gamma$</td>
<td>'1'</td>
<td>$\sigma_{\mu\nu}0 + 2$</td>
</tr>
<tr>
<td></td>
<td>$\kappa + 2 = \gamma$</td>
<td>'1'</td>
<td>$\sigma_{\mu\nu}0 + 2$</td>
</tr>
<tr>
<td></td>
<td>$\kappa + 2 &gt; \gamma$</td>
<td>'1'</td>
<td>$\sigma_{\mu\nu}0 + 2$</td>
</tr>
<tr>
<td>$\sigma_{\mu\nu}0 + 2 = \gamma$</td>
<td>$\kappa + 2 &lt; \gamma$</td>
<td>'2'</td>
<td>$\kappa + 2$</td>
</tr>
<tr>
<td></td>
<td>$\kappa + 2 = \gamma$</td>
<td>$D$</td>
<td>$\gamma$</td>
</tr>
<tr>
<td></td>
<td>$\kappa + 2 &gt; \gamma$</td>
<td>$T_2$</td>
<td>$p &gt; \gamma$</td>
</tr>
<tr>
<td>$\sigma_{\mu\nu}0 + 2 &gt; \gamma$</td>
<td>$\kappa + 2 &lt; \gamma$</td>
<td>'2'</td>
<td>$\kappa + 2$</td>
</tr>
<tr>
<td></td>
<td>$\kappa + 2 = \gamma$</td>
<td>$T_1$</td>
<td>$p &gt; \gamma$</td>
</tr>
<tr>
<td></td>
<td>$\gamma &lt; \kappa + 2 &lt; \sigma_{\mu\nu}0 + 2$</td>
<td>'3'</td>
<td>$\gamma$</td>
</tr>
<tr>
<td></td>
<td>$\kappa + 2 = \sigma_{\mu\nu}0 + 2$</td>
<td>'3'</td>
<td>$\gamma$</td>
</tr>
<tr>
<td></td>
<td>$\kappa + 2 &gt; \sigma_{\mu\nu}0 + 2$</td>
<td>'3'</td>
<td>$\gamma$</td>
</tr>
</tbody>
</table>
We can, to some extent, control the values of $\gamma$ and of $\sigma_{\mu\nu0}$; but we have no control over $\kappa$, since $H$ is a variable. Table 3 (based on Figure 4 and Table 2) shows the dependence of $p_{\mu\nu H}$ on all three parameters, for all their possible relative magnitudes. The 'bad' (high-consonance) cases arise in the triangular plane regions $T_1$, $T_2$, and $T_3$; and least damage is done if bad values of $H$ are as few as possible. Note that half the values of $H$ are odd ($\kappa = 0$), a quarter are divisible by 2 but not by 4 ($\kappa = 1$), an eighth are divisible by 4 but not by 8 ($\kappa = 2$), etc.; with the value $\kappa = \eta$, say, accounting for a fraction $2^{-\eta-1}$ of all values of $H$; and with all values of $\kappa > \eta$ accounting for the same fraction. Thus, if $\sigma_{\mu\nu0} + 2 > \gamma$, the fraction of bad $H$-values (in $T_1$: $\kappa = \gamma - 2$) is $2^{-\gamma+1}$; if $\sigma_{\mu\nu0} + 2 = \gamma$, the fraction (in $T_2$: $\kappa > \gamma - 2$) is again $2^{-\gamma+1}$; and if $\sigma_{\mu\nu0} + 2 < \gamma$, the fraction (in $T_3$: $\kappa = \sigma_{\mu\nu0}$) is $2^{-\sigma_{\mu\nu0}-1} > 2^{-\gamma+1}$. We therefore see that it is desirable to take $\Delta_{\mu\nu1}$ odd $[\sigma_{\mu\nu1} = 0]$ and $\gamma \geq 3$ [as noted in (108)], and then

$$\sigma_{\mu\nu0} \geq \gamma - 2.$$  \hspace{1cm} (117)

Since the fraction of 'bad' values of $H$ is then $2^{-\gamma+1}$, it is probably wise to exceed the criterion in (108) somewhat, to make this fraction smaller. A reasonable condition might be

$$\gamma \geq 8.$$  \hspace{1cm} (118)

yielding a fraction $2^{-7}$ (less than 1%) of bad values of $H$. This is achieved, for example, by taking every $b_\mu \equiv 1$ (mod 256). As the lower bound on $\gamma$ increases, (a) the 'good' values of $H$ yield somewhat less desirable consonances, and (b) the numbers of available distinct values of $b_\mu$ and of $x_{\mu0}$ decrease correspondingly; so there is a trade-off here, as in so many such situations, and an 'engineering solution' (i.e., a compromise) is indicated.

Note the special solution, when

$$\Delta_{\mu\nu0} = 0; \text{ i.e., } \sigma_{\mu\nu0} = \infty.$$  \hspace{1cm} (119)

Then, as is pointed out in Corollary 4, we have $p_{\mu\nu H} = \gamma$ for all $H$; but at the cost of no choice of $\sigma_{\mu\nu0}$ and so of $x_{\mu0}$.

We must not overemphasize the importance of the consonances of positionally offset pairs of sequences. The unfortunate results can, to some extent, be minimized by suitably avoiding unfavorable offsets;
but only at the cost of wasting some random numbers which might otherwise be put to use. Again, suitable compromises are indicated.

Finally, we consider the consonance between sequences not arising from the same branch-point. Of course, Theorem 2 still applies. Let \( \Phi_\mu \) and \( \Phi_\nu \) begin at nodes numbered \( N_\mu = 2\mu + 1 \) and \( N_\nu = 2\nu + 1 \), respectively, but now without the relation (88). Then the level, \( m \), of \( N_\mu \) is given by

\[
2^{m-1} \leq \mu \leq 2^m - 1; \quad \text{i.e.,} \quad m = \lceil \log_2(\mu + 1) \rceil, \tag{120}
\]

and the level, \( n \), of \( N_\nu \) will be determined similarly. Now, \( \beta \) and \( \gamma \) will still be defined by (101) and (104); but the appropriate \( \delta_0 \) [see (26)] will now be

\[
\delta_0 = x_{\mu(k-m)} - x_{\nu(k-n)}, \quad \text{where} \quad k = \max(m, n). \tag{121}
\]

Whatever condition we apply to all the \( b_\eta \), to ensure (108) or (118), will still help us here; but all the \( x_{\eta 0} \) will already have been fixed (as discussed above) in a way that will not likely help us here. The new \( \delta_0 \) and \( \sigma \) will thus be out of our control; whence the consonance \( 2^p \) of \( \Phi_\mu \) and \( \Phi_\nu \) will float freely, in accordance to Theorem 2 and Figure 3, with \( p \leq \gamma \), except for the 'bad' cases, when \( \sigma = \gamma - 2 \). As before, this will tend to occur about \( 2^{-\gamma+1} \) of the time.

5. SPECIFIC PROCEDURES

We now have all the underlying machinery that we shall need, to select specific procedures, to generate tree-structured families of linear congruential pseudo-random generators, yielding sequences which are individually uniform, with minimal coarseness, and which are mutually independent, with acceptably low consonances.

To put things in perspective, we observe that, for a given fixed choice of the parameter \( a \) [which we have supposed to satisfy (95)], there are \( 2^{M-3} \) distinct possible values of the \( b_\eta \) satisfying (108) [or \( 2^{M-8} \) distinct values satisfying (118)], and altogether \( 2^M \) distinct possible values of the \( x_{\eta 0} \). The possible distinct pseudo-random
sequences are thus in any case no more than \(2^{2M-3}\) in number; and probably less, in any given procedure (e.g., in Warnock's algorithm, there are only at most \(2^M\) distinct sequences). Since the sequences begin at all the odd-numbered nodes (numbered \(N_\mu = 2\mu + 1\)) of a binary tree [see Figure 1], it is clear that there must be at least one repetition in the first \(2M - 1\) levels, and thereafter, more and more frequently within each level (since Level \(2M - 2\) alone has \(2^{2M-2}\) nodes, and so \(2^{2M-3}\) odd-numbered nodes; and each level has twice as many nodes as its immediate predecessor). We thus cannot expect to avoid the recurrence of the same pseudo-random sequences at scattered points in our binary tree. (Even if we were to exploit every possible sequence of the form (4) in our tree, there would still have to be at least one repetition in the first \(3M + 2\) levels.) In practice, it is extremely difficult to avoid the occurrence of repetitions somewhat more frequent than these extreme bounds. However, we must recall that the nodes of our binary tree correspond to batches of \(T\) consecutive pseudo-random numbers [see Lemma 18], one of which usually suffices to generate a single physical event; and these events will rarely lead to actual branching (or, as has been pointed out, the resulting computations would be enormously, impossibly, too laborious). Thus only a very sparse, random sample of the branches is actually exploited in any realistic calculation. This is what saves us, in practice. Nevertheless, any repetitions that do occur must be minimized with respect to quantity, and dispersed as far as possible in their distribution over the tree.

Perhaps the simplest hypothesis to adopt would be that
\[
x_{\nu, 0} = \sum_{\mu} (2^m N_\mu b_\mu x_{\mu m}) = x_{\mu n}.
\]

(122)

This is comparable in economy to Warnock's definition of \(B_{\nu r}\) in (96), and is equivalent, of course, by (106), to (119).

Note that all the parameters \(b_\nu\) are postulated to be odd, with (108) holding; which we can ensure by choosing, once and for all, any odd value \(b_0 = 2\theta + 1\), and then taking
\[
(\forall \nu \geq 0) \quad b_\nu \equiv b_0 \pmod{8}.
\]

(123)

Since every starting node of a new generator \(\Phi_\nu\) has an odd number, \(N_\nu = 2\nu + 1\), with all the \(\nu\) different, of course; it is natural to adopt the
simple relation
\[ b_v = B_{N,3}(2^m N^\mu, b^\mu, x_\mu m) = <8v + b_0 | \Omega >. \]  
(124)

As a slight generalization, we may consider
\[ b_v = B_{N,\phi}(2^m N^\mu, b^\mu, x_\mu m) = <2^\phi v + b_0 | \Omega >, \]  
(125)

where
\[ b_0 = <2\theta + 1 | 2^\phi > \]  
(126)

and [see (75) and (108)]
\[ M \geq \phi \geq 3. \]  
(127)

Since \( b_v - b_0 \in J_\phi \) [see (69)], we see that there will be exactly \( 2^{M-\phi} \)
distinct values of \( b_v \) satisfying (125). Obviously, for (125),
\[ \beta_{\mu v} = <2^\phi (\mu - v) | \Omega >; \]  
(128)

whence, \( \gamma \geq \phi \). This result would indicate that, in fact, (124), with \( \phi = 3 \), is the best choice; though the considerations leading to (118) would suggest something closer to \( \phi = 8 \), instead.

Observe that, while Warnock has a single generator family identified by the parameters \((a, b_v)\) for all left-slanting sequences [see (96)], and generates \( x_{v0} \) from \( x_{\mu m} \) by means of another single generator \((a^\dagger, b^\dagger)\); we propose to have the \( b_v \) specified from the index \( v \) by a formula [see (125)], and \( x_{v0} \) equal to the parent value \( x_{\mu m} \).

If the parameters \( a, b^\mu, \) and \( b_v \) satisfy (95) and (123), and we make \( \Lambda_{\mu v0} = <x_{\mu m} - x_{v0} | \Omega > \) even; then, by (108) and (113). Corollary 5(iii) applies, and the consonance between 'parallel' sequences, \( [x_{\mu (t+m+1)}]_{t=0}^\infty \), beginning at node \( 2v = 2^m+1(2^\mu + 1) \), and \( [x_{v'}]_{j=0}^\infty \), beginning at node \( 2v + 1 \), will be \( 2^{2^\mu+1} = 2^2 = 4 \), which is just fine.

If we adopt the simple algorithm embodied by (122), then the node record \( R_v \) [see (89)] suffices to carry all necessary information at every node, for initializing a right-slanting branch whenever needed.
Algorithm 1. The procedure carries at each node, numbered \( v = 2^m N_{\mu} = 2^m (2\mu + 1) \), a record \( R_v = (2^m N_{\mu}, b_{\mu}, x_{\mu m}) \), the transformation for which, on passage to the two child-nodes is given by

\[
R_{2v} = L_{S,\phi}(R_v), \quad R_{2v+1} = R_{N_v} = M_{S,\phi}(R_v). \tag{129}
\]

These mappings are defined by

\[
L_{S,\phi}(R_v) = \begin{cases} 
2v = 2^{m+1} N_{\mu} \\ \text{b-value at } 2v \\ x_{\mu(m+1)}
\end{cases} \leftrightarrow \begin{cases} 
2 \times (v = 2^m N_{\mu}) \\ \text{(b-value at } v) = b_{\mu} \\ \langle ax_{\mu m} + b_{\mu} \rangle | Q\rangle
\end{cases} \tag{130}
\]

and

\[
M_{S,\phi}(R_v) = \begin{cases} 
2v + 1 = 2^{m+1} N_{\mu} + 1 \\ \text{(b-value at } N_v) = b_v \\ x_{v0}
\end{cases} \leftrightarrow \begin{cases} 
2 \times (v = 2^m N_{\mu}) + 1 \\ \langle 2^\phi v + b_0 \rangle | Q\rangle \\ x_{\mu m}
\end{cases}. \tag{131}
\]

This agrees with (122) and (125). Thus, \( x_{v0} = X_S(R_v) \) and \( b_v = B_{N,\phi}(R_v) \).

However, as has been borne out by some ingeniously contrived, but realistically possible, simulations performed by T. E. BOOTH [private communication], there can well occur many identical replications of sequences. This undesirable situation may not show up, because of the extreme sparseness of the subtree actually occurring in any practical computation; but the possibility nevertheless remains and presents a serious, lurking threat. It is to reduce this risk that Algorithm 2 is developed below.

We have seen that, for a fixed value of the parameter \( a \), there are at the very most \( 2^{2M-3} \) distinct linear-congruential sequences; so that a repetition must occur within at most \( 2M - 1 \) levels. We now seek to construct a scheme which will guarantee the absence of all repetitions for as many levels as possible.

The class of generation schemes which we shall henceforth consider has \( b \)-values determined by (125) with a suitable choice of \( \phi \).
satisfying (127). A b-value will be said to originate (or to have its origin) at an odd-numbered node; thereafter, it will apply to all the (even-numbered) left-slanting descendants of this node.

**Definition 6.** Given a binary tree (see Figure 1) associated with a family \( \Phi = \Phi(\mu = 0, 1, 2, \ldots) \) of linear-congruential generators [with fixed parameter \( a \) satisfying (95), and all the \( b_\mu \) odd and satisfying (125) for fixed \( \phi \)]; we define the first \( k + 1 \) levels of the tree (i.e., Levels 0 through \( k \)) to be the \( k \)-body of the tree and we denote it by \( B_k \). In particular, we define the \( (M - \phi) \)-body of the tree (i.e., the first \( M - \phi + 1 \) levels of the tree; Levels 0 through \( M - \phi \)) to be the apex of the tree and we denote it by \( B_{M-\phi} = A_\phi \).

**Lemma 20.** Any particular b-value, say \( b^* \), will occur at intervals of length \( 2^{M-\phi} \) in the index \( v \) (this correspond to intervals of length \( 2^{M-\phi+1} + 1 \) in the node-number \( 2v + 1 \)); and will therefore originate once and only once in the apex \( A_\phi \) of the binary tree specified in Definition 6. For any

\[
k > M - \phi,
\]

the \( k \)-body \( B_k \) of the tree will contain exactly \( 2^{k-M+\phi} \) origins of the b-value \( b^* \); with one origination in the apex, one in Level \( M - \phi + 1 \), two in Level \( M - \phi + 2 \), four in Level \( M - \phi + 3 \), \ldots, and \( 2^{k-M+\phi-1} \) in Level \( k \).

\(<<\) By (85) and (128) [which is a consequence of (125)], whatever odd b-value \( b_0 \) [see (126)] we choose for the root of the tree, every \( \beta_{\mu v} = \langle b_\mu - b_v | Q \rangle = \langle 2^\phi (\mu - v) | Q \rangle \), and, therefore, \( b_\mu = b_v \) if and only if \( 2^{M-\phi} | (\mu - v) \); so that any b-value \( b^* \) repeats at intervals of length \( 2^{M-\phi} \) in the index \( v \). Since the indices occurring in the apex \( A_\phi \) of the tree are 0, 1, 2, \ldots, \( 2^{M-\phi} - 1 \) \( (2^{M-\phi} \) consecutive, distinct values; corresponding to the consecutive odd node-numbers, 1, 3, 5, \ldots, \( 2^{M-\phi+1} - 1 \)], these indices are all less than \( 2^{M-\phi} \) apart; and therefore no b-value can originate twice in the apex. Since there are, altogether, \( 2^{M-\phi} \) possible b-values satisfying (125), each possible value must originate just once in \( A_\phi \).

Figure 1 illustrates the fact that, for any \( h > 0 \), Level \( h \) of a binary tree contains \( 2^h \) consecutively-numbered nodes, half of which, \( 2^{h-1} \), are odd-numbered. Therefore, the \( k \)-body \( B_k \) of the tree will contain \( 2^{k+1} - 1 \) consecutively-numbered nodes: \( 2^k - 1 \) of them even-numbered and \( 2^k \) of them odd-numbered; the latter corresponding to
Indices 0, 1, 2, \ldots, 2^k - 1. Since we have just shown that each possible \( b \)-value \( b^* \) originates at intervals of just \( 2^{M-\phi} \) in the index, it follows that each \( b^* \) will originate just once in the apex: exactly \( 2^{h-M+\phi-1} \) times in Level \( h \) (for \( h = M - \phi + 1, M - \phi + 2, M - \phi + 3, \ldots, k \)); and just \( 2^{k-M+\phi} \) times, altogether, in the \( k \)-body of the tree.

We shall henceforth further restrict the class of generation schemes considered, to those in which, at any odd-numbered node \( N_v \), first, the \( b \)-value \( b^* \) is determined by (125), then a tentative initial \( x \)-value \( x^* \) is obtained in some computationally efficient way, its parity is compared with that of the \( x \)-value \( x_\mu \) at the parent-node \( v \), and, only if the parities differ, \( x^* \) is replaced by its successor \( \langle ax^* + b^* \rangle \) in the sequence, to yield the actual initial \( x \)-value \( x_0 \); thus we impose the parity-condition, that \( \Delta_{\mu v}^0 \) should be even [see (106), (113), and Corollary 5].

**Definition 7.** If we select an \( a \in J \) satisfying (95), an odd \( b_0 \in J \), an integer \( f_0 \in J \), an integer \( \phi \) satisfing (127), an integer \( \psi \) such that \( \phi < \psi \leq M \), and a non-negative index \( v \); we can uniquely define [see (125); use \( \psi > \phi \) in (135)]

\[
v_0 = \langle v \rangle 2^{M-\phi}, \quad s = (v - v_0)/2^{M-\phi};
\]

whence

\[
b^* = b_v = \langle 2^\phi v + b_0 \rangle = \langle 2^\phi v_0 + b_0 \rangle.
\]

and

\[
x^* = \langle 2^\psi v + f_0 \rangle = \langle 2^\psi v_0 + f_0 \rangle.
\]

(Note that \( b^* \) and \( x^* \) are clearly functions only of \( v_0 \), not of \( s \).) Then the sequence \( [x_j^*]_{j=0}^\infty \), with generator \( \Phi^* = \Phi(a, b^*, x^*) \) will be called the master sequence belonging to \( b^* \).

**Lemma 21.** Any sequence \( [x_j]_{j=0}^\infty \), with parameters \( a, b^* \) and initial \( x \)-value \( x_0 \); i.e., with generator \( \Phi = \Phi(a, b^*, x_0) \), will be a displacement, along its length, of the master sequence belonging to the given \( b^* \).

\( \langle \langle \text{Since } a \text{ satisfies (95) and } b^* \text{ is odd, we are in Case (i) of Lemma 12 } \{c = 0, \text{ by (40), } d \geq 2, \text{ by (52), and } s \geq 0, \text{ by (41)} \}\rangle \text{ so that the master sequence [all of whose members are in } J, \text{ by (4) and (28)] has period } Q. \text{ By Lemma 17, this means that the master sequence} \rangle \)
runs through exactly $Q$ distinct values. Since every $x_0 \in J$, whose cardinality is $|J| = Q$ also, it follows that all these $x_0$ are members of the master sequence. Therefore any generator $\Phi$ generates the master sequence, displaced to the member $x_0$ as its starting-point.

**Figure 5**

Sequences with a common $b$-value

Analysis of $x$-value counts in sequences (left-sloping chains) arising from the same $b$-value $b^*$, down to Level $k$. The $b$-value repeats every $2^{M-\phi}$ index-values; so originates once in the "apex", once in Level $M-\phi+1$, twice in Level $M-\phi+2$, and so on. $T_s$ is the cumulative number of nodes to be skipped, in the "master sequence", beginning at node $2v_0 + 1$ in the apex, to the beginning the s-th sequence, at node $2v_s + 1 = 2v_0 + 1 + s2^{M-\phi}$. The $s$-values are given in the circles representing the initial nodes.
Lemma 22. For each possible odd value $b^*$ given by (134), the total number of $x$-values generated in all the sequences occurring in the $k$-body $B_k$, with parameters $(a, b^*)$, does not exceed

$$Z_k = M - \phi - 2 + 3 \times 2^{k-M+\phi}.$$  \hspace{1cm} (136)

The situation is illustrated in Figure 5. By Lemma 20, the value $b^*$ will originate just once, with some index $v_0 < 2^{M-\phi}$, in the apex $A_\phi$, at some level ranging from 0 to $M - \phi$. The tentative initial $x$-value $x_{v_0}^*$ will then be computed, and the parity-check may yield a skip-forward in the master sequence; except if we are at the root (when $v = 0$), for then no check is necessary or possible, and therefore no such parity-skip can occur. If the resulting sequence originates at Level $h$ ($0 \leq h \leq M - \phi$, since we are in the apex), it will generate exactly $(k - h + 1)$ $x$-values, one at each level of the $k$-body $B_k$, beginning with Level $h$; and so the number $r_k$ of $x$-values needed by this sequence will not exceed $k + 1$ (it will be $k - h + 1 = k + 1$, if $h = 0$; and at most $k - h + 2 \leq k + 1$, if $h \geq 1$).

The single sequence which, Lemma 20 tells us, will originate in Level $M - \phi + 1$ with the $b$-value $b^*$, will similarly generate no more than $(k - M + \phi + 1)$ $x$-values, allowing for a possible parity-skip; the two sequences originating in Level $M - \phi + 2$ will, together, generate no more than $2(k - M + \phi)$ $x$-values, since they start one level lower; and so on. Thus, the total number of $x$-values generated in Levels 0 through $k$, by sequences having the $b$-value $b^*$ will not exceed

$$Z_k = k + 1 + (k - M + \phi + 1) + 2(k - M + \phi) + 4(k - M + \phi - 1)$$

$$+ \ldots + 2^{h-M+\phi-1} (k - h + 2) + \ldots + 2^{k-M+\phi-1} \times 2$$

$$= \{ k + 1 \}_{1} + [2(k - M + \phi + 1) - \{ k - M + \phi + 1 \}_{1}]$$

$$+ [4(k - M + \phi) - 2(k - M + \phi)]$$

$$+ [8(k - M + \phi - 1) - 4(k - M + \phi - 1)]$$

$$+ \ldots$$

$$+ [\{ 2^{k-M+\phi} \times 2 \}_{2} - 2^{k-M+\phi-1} \times 2].$$

The middle expression above partially 'telescopes' (excepting the terms in $\{ \ldots \}$, which are combined according to the small subscripts attached thereto) to yield
\[ Z_k = \{ M - \phi \}_1 + 2 + 4 + 8 + \ldots + 2^k - M + \phi - 1 + \{ 2^k - M + \phi + 1 \}_2 \]

\[ = M - \phi - 2 + 2^k - M + \phi + 2^k - M + \phi + 1, \]

which is (136).

Let us write

\[ M' = M - \phi \quad \text{and} \quad k' = k - M' = k - M + \phi; \quad (137) \]

then (127) and (132) become

\[ \phi \geq 3, \quad M' \geq 0, \quad \text{and} \quad k' \geq 1; \quad (138) \]

and (136) becomes

\[ Z_k = M' - 2 + 3 \times 2^{k'} \quad (139) \]

Lemma 21 tells us that all sequences of x-values with parameters \( a \) and \( b^* \) will be displacements of the \((a, b^*)\) master sequence. Such sequences will intersect \( B_k \) [in the sense that each node of \( B_k \) carries an x-value] in segments (i.e., continuous sequential pieces) of this master sequence. In order to avoid repetitions of x-values in these segments, it will be necessary, for all of them to be disjoint; so that, for each fixed value of \( b^* \), all \( Z_k \) x-values counted in Lemma 22 will have to be different. Of course, this cannot be done for arbitrarily large \( k \); so we shall have to find an upper bound on \( k \) for which a solution exists. As was argued in the proof of Lemma 21, any master sequence has just \( Q = 2^M \) x-values in it, and \( Q \) is the total number of possible x-values, by (28). Thus, for feasibility, we require that

\[ Z_k \leq 2^M; \quad (140) \]

by (137) and (139), this means that

\[ M' - 2 + 3 \times 2^{k'} \leq 2^M - \phi. \quad (141) \]

**Lemma 23.** Subject to the condition (138), the inequality (141) is satisfied if and only if

\[ k' \leq M' + \phi - 2. \quad (142) \]

\[ \ll \ll \text{Note that, by (138), } M' + \phi - 2 \geq 1 \text{ and } k' \geq 1; \text{ so that it is possible for both (138) and (142) to hold true.} \]
Now consider the inequality
\[ M' - 2 < 2^{M'+1}. \]  
It is certainly true when \( 0 \leq M' \leq 2 \) (since the left-hand side is then negative or zero, while the right-hand side is positive). Suppose, therefore, that (143) holds for \( M' = t \geq 0 \) (so that \( 2^{t+1} \geq 2^1 = 1 \)); then \( t - 2 < 2^{t+1} \); whence
\[
(t + 1) - 2 = (t - 2) + 1 < 2^{t+1} + 1 < 2^{t+1} + 2^{t+1} = 2^{(t+1)+1};
\]
that is, (143) holds for \( M' = t + 1 \); and so, by induction on \( t \), (143) is proved for all \( M' \). Since, by (138), \( \phi - 2 \geq 1 \), it follows from (143) that
\[ M' - 2 < 2^{M+\phi-2}. \]  
(144)

If (142) holds, then, by (144),
\[ M' - 2 + 3 \times 2^{k'} < M' - 2 + 3 \times 2^{M+\phi-2} < 4 \times 2^{M+\phi-2} = 2^{M+\phi}, \]
which is the required inequality (141).

Contrariwise, if (142) does not hold, then, since we are dealing in integers,
\[ k' \geq M' + \phi - 1. \]  
(145)

Now consider the inequality
\[ 2 - M' < 2^{M+2}. \]  
(146)
It is certainly true whenever \( M' \geq 2 \) (since the left-hand side is then negative or zero, while the right-hand side is positive). Since, by (138), \( M' \geq 0 \); this leaves \( M' = 0 \), when (146) is \( 2 < 2^2 \); and \( M' = 1 \), when it is \( 1 < 2^3 \). Thus, (146) is true for all \( M' \geq 0 \). Now, by (138), \( \phi - 1 \geq 2 \); so, by (145) and (146),
\[ M' - 2 + 3 \times 2^{k'} \geq (M' - 2 + 2^{M+\phi-1}) + 2 \times 2^{M+\phi-1} \]
\[ \geq (M' - 2 + 2^{M+2}) + 2 \times 2^{M+\phi-1} \]
\[ > 2 \times 2^{M+\phi-1} = 2^{M+\phi}, \]
which contradicts (141). This completes the proof of the lemma.>>
The inequalities (138) and (142) are illustrated in Figure 6. Since we seek to maximize the height of the $k$-body $B_k$, so as to have as many segments as possible disjoint, and for the greatest possible length, it is clear from Lemma 23 that we must select

$$k' = M' + \phi - 2, \quad \text{i.e.,} \quad k = 2M - \phi - 2. \quad (147)$$

![Figure 6](image)

**Figure 6**

Graph of allowed $(M', k')$ region

$M' = M - \phi$ and $k' = k - M + \phi$ satisfy the inequalities (138) and (142). This makes allowable the region shaded in the figure. Since $M$ and $\phi$ are given first, and we seek the greatest possible $k$, it is clear that the sloping line, $k' = M + \phi - 2$, or $k = 2M - \phi - 2$ yields the best value of $k$. 
For any index $v$ occurring in $B_k$, the node $N_v$ is, by (133), in the apex $A_\phi$ (see proof of Lemma 20); the $b$-value $b_v = b^*$, given by (134), will originate at the odd-numbered nodes $N_v$ with indices denoted by

$$ v_0, \quad v_1 = v_0 + 2 M - \phi, \quad v_2 = v_0 + 2 	imes 2 M - \phi, $$

$$ \ldots, \quad v_t = v_0 + t \times 2 M - \phi, \ldots; \quad (148) $$

and, in particular, by (133), $v = v_s$. Node $N_{vt}$ will be in $A_\phi$, for $t = 0$; in Level $M - \phi + 1$, for $t = 1$; in Level $M - \phi + 2$, for $t = 2$ and 3; and so on. This node will be in Level $h$, if $h > M - \phi$, for

$$ t = 2^h - M - \phi - 1, \quad 2^h - M - \phi + 1, \quad 2^h - M - \phi - 1 + 2, \quad \ldots, \quad 2^h - M - \phi - 1. $$

If $v < 2^M - \phi$ ($s = 0$), the node $N_{vs} = N_v$ will be in $A_\phi$; otherwise, it will be in Level $h$, with $h > M - \phi$, if

$$ 2^h - M - \phi - 1 \leq s < 2^h - M - \phi. \quad (149) $$

We can express the index $v$ in binary notation as

$$ v = \{B_{k-1} \ldots B_{M-\phi} B_{M-\phi-1} \ldots B_0\}; \quad (150) $$

where the $B_j$ are the uniquely determined bits of $v$ (binary digits; taking the value 0 or 1), given by

$$ B_j = \lfloor <v|2^{h+1}>/2 \rfloor \quad (151) $$

Here, $\lfloor \ldots \rfloor$ denotes the “floor” function (the integer infimum), just as $\lceil \ldots \rceil$ denotes the “roof” function (the integer supremum); for example, $\lfloor 17 \rfloor = \lceil 17 \rceil = 17$, but $\lfloor 28.3 \rfloor = 28$ and $\lceil 28.3 \rceil = 29$. Since there are just $2^k$ odd-numbered nodes (indexed from 0 through $2^k - 1$) in Levels 0 through $k$, the $k$ bits shown in (150) suffice for any index occurring in the $k$-body $B_k$.

By (133), $s = \{B_{k-1} \ldots B_{M-\phi}\}$; and

$$ v_0 = \{B_{M-\phi-1} \ldots B_0\} = \{Y B_{M-\phi-2} \ldots B_0\}. \quad (152) $$

where

$$ Y = B_{M-\phi-1}. \quad (153) $$

Define $n_s$ by

$$ \begin{cases} n_s = 0 & \text{if } s = 0 \\ 2n_s - 1 \leq s < 2n_s & \text{if } s \geq 1 \end{cases} \quad (154) $$
Then, clearly, \( n_s \) is the number of significant bits in \( s \); that is,

\[
B_{k-1} = \ldots = B_{M-\phi+n_s} = 0 \quad \text{and} \quad B_{M-\phi+n_s-1} = 1;
\]

and so

\[
s = [0 \ 0 \ldots \ 0 \ 1 \ B_{M-\phi+n_s-2} \ldots \ B_{M-\phi}]. \quad (155)
\]

Observe that, if \( s > 0 \) (i.e., if node \( N_v \) is not in \( A_\phi \); or, equivalently, \( v \geq 2^M-\phi \)); then, for any node \( N_v \) in Level \( h > M - \phi \) [compare (149) with (154)],

\[
n_s = h - M + \phi > 0. \quad (156)
\]

In other words, all nodes \( N_{v_s} \) with given \( n_s > 0 \) (and varying \( v_0 \) and \( s \)) are in the same level, \( h_s \), and

\[
\text{if } n_s > 0 \text{ then } h_s = M - \phi + n_s. \quad (157)
\]

If we pass from the parent-node, numbered \( v \), to its left and right child-nodes, numbered \( 2v \) and \( N_v = N_{v_s} = 2v + 1 \), the node-numbers change from

\[
[0 \ldots 0 \ 0 \ 1 \ B_{M-\phi+n_s-2} \ldots \ B_{M-\phi} \ Y B_{M-\phi-2} \ldots B_0]
\]

into

\[
[0 \ldots 0 \ 1 \ B_{M-\phi+n_s-2} \ldots \ B_{M-\phi} \ Y \ B_{M-\phi-2} \ldots B_0 \ 0]; \quad (158)
\]

and

\[
[0 \ldots 0 \ 1 \ B_{M-\phi+n_s-2} \ldots \ B_{M-\phi} \ Y \ B_{M-\phi-2} \ldots B_0 \ 1]
\]

where the diamond (\( \diamond \)) marks the separation between the bits of \( s \) and those of \( v_0 \). Therefore, if the node \( N_v \) is in the apex (\( s = 0 \)) and if further, \( Y = B_{M-\phi-1} = 0 \), then the nodes \( N_{2v} \) and \( N_{2v+1} \) are also in the apex, and neither \( s \) nor \( n_s \) will change; while, if the nodes \( N_{2v} \) and \( N_{2v+1} \) are not in the apex (\( s = 0 \) and \( Y = 1 \), or \( s \geq 1 \); i.e., \( v \geq 2^{m-\phi-1} \)), then \( n_s \) will increase by just 1 and \( s \) will become \( 2s + Y \). If we denote the values of \( s \) and \( n_s \) for \( N_{2v} \) by \( s' \) and \( n'_s \), and for \( N_{2v+1} \) by \( s'' \) and \( n''_s \), then

\[
s' = s'' = 2s + Y
\]

\[
n'_s = n''_s = \begin{cases} n_s + Y, & \text{if } s = 0 \\ n_s + 1, & \text{if } s \geq 1 \end{cases}
\]

When we wish to branch to the right, it is a practical necessity to do so without information about the many previous pseudo-random
numbers generated in the current calculation. In the proof of Lemma 22, we determined an upper bound for the maximum number of $x$-values that may need to be computed for each of the segments, of the master sequence belonging to any selected $b$-value $b^*$, occurring in the $k$-body $B_k$ as parts of sequences with parameters $(a, b^*)$. We use this information to guarantee the disjointness of all these segments.

To go to a right branch from a node $v$, we shall proceed as follows:

1. Carry, in the current node-record, or quickly compute, $v_0, s, b^*$, and $x^*$ [see (133) - (135)].

2. Compute the tentative initial $x$-value $x_v^*$, by its displacement $T_s$ from the initial $x$-value $x^*$ along the master sequence. Use the notation [similar to (77)]

$$A_s = \langle a^{T_s} | G \rangle, \quad S_s = \langle S_{T_s}(a) | G \rangle, \quad B_s = \langle S_{s} b^* | G \rangle, \quad (160)$$

and apply (30) to yield that

$$x_v^* = \langle A_s x^* + B_s | G \rangle.$$  \hspace{1cm} (161)

3. Carry, in the node-record, the current $x$-value $x_{\mu m}$.

4. Compare the parities of $x_{\mu m}$ and $x_v^*$; if they are the same, take the initial $x$-value of the new sequence to be $x_{v0} = x_v^*$; if the parities differ, take $x_{v0} = \langle ax_v^* + b^* \rangle$.

Following the proof of Lemma 22, we choose the displacements $T_s$ in such a manner as to allow enough space, along the master sequence, for the maximum number of $x$-values that may be needed by the segments of sequences originating at earlier nodes ($N_{v_0}, N_{v_1}, N_{v_2}, \ldots, N_{v_{s-1}}$). The situation is sketched in Figure 5, and the results are tabulated in Table 4. As the simplest choice, we take

$$T_0 = 0, \quad \text{i.e.,} \quad x_0^* = x^*.$$  \hspace{1cm} (162)

We see from Table 4 that the rule, for $s \geq 1$, is

$$T_{s+1} - T_s = k - M + \phi + 2 - n_s.$$  \hspace{1cm} (163)

with $n_s$ defined in (154). For $k$ optimal [see (147)], this gives that

$$T_{s+1} - T_s = M - n_s;$$  \hspace{1cm} (164)

whence, for $s \geq 2$, since $T_1 = k + 1 = 2M - \phi - 1$. 

- 54 -
We observe that, for any integer \( n > 0 \), \( n_r = n \), when \( r = 2^n - 1, \ 2^n - 1 + 1, \ 2^n - 1 + 2, \ldots, 2^n - 1 \) (i.e., for \( 2^n - 1 \) consecutive values of \( r \)). Thus,

\[
T_s = T_1 + \sum_{r=1}^{s-1} (T_{r+1} - T_r) = T_1 + \sum_{r=1}^{s} (M - n_r) - M + n_s
\]

\[
= (s + 1)M + n_s - \phi - 1 - \sum_{r=1}^{s} n_r. \tag{165}
\]

<table>
<thead>
<tr>
<th>( s )</th>
<th>( n_s )</th>
<th>LEVEL OF ( N_{vs} ) (CURRENT NODE)</th>
<th>LEVEL OF ( N_{vs+1} ) (NEXT NODE)</th>
<th>DISPLACEMENT ((T_{s+1} - T_s))</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>in Apex</td>
<td>1</td>
<td>( k + 1 )</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>( k - M + \phi + 1 )</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>( k - M + \phi )</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>( k - M + \phi )</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>( k - M + \phi - 1 )</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>( k - M + \phi - 1 )</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>( k - M + \phi - 1 )</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>( k - M + \phi - 1 )</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>( k - M + \phi - 2 )</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>( k - M + \phi - 2 )</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>( k - M + \phi - 2 )</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>( k - M + \phi - 2 )</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>( k - M + \phi - 2 )</td>
</tr>
<tr>
<td>13</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>( k - M + \phi - 2 )</td>
</tr>
<tr>
<td>14</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>( k - M + \phi - 2 )</td>
</tr>
<tr>
<td>15</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>( k - M + \phi - 2 )</td>
</tr>
<tr>
<td>16</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>( k - M + \phi - 3 )</td>
</tr>
<tr>
<td>17</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>( k - M + \phi - 3 )</td>
</tr>
<tr>
<td>18</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>( k - M + \phi - 3 )</td>
</tr>
</tbody>
</table>
\[ \sum_{r=1}^{s} n_r = 1 + 2 \cdot 2 + 2^2 \cdot 3 + 2^3 \cdot 4 + \ldots + 2^{n_s-2} (n_s - 1) + n_s (s + 1 - 2^{n_s-1}) \]

\[
\begin{align*}
= & \quad 2 \times 1 - 1 \times 1 \\
+ & \quad 2^2 \times 2 - 2 \times 2 \\
+ & \quad 2^3 \times 3 - 2^2 \times 3 \\
+ & \quad 2^4 \times 4 - 2^3 \times 4 \\
+ & \quad \ldots \\
+ & \quad \{2^{n_s-1} (n_s - 1)\}_1 - 2^{n_s-2} (n_s - 1) + \{n_s (s + 1 - 2^{n_s-1})\}_2 \\
= & \quad \{2^{n_s-1} (n_s - 1)\}_1 + \{n_s (s + 1 - 2^{n_s-1})\}_2 \\
- & \quad (1 + 2 + 2^2 + \ldots + 2^{n_s-2}) \\
= & \quad n_s (s + 1) - (1 + 2 + 2^2 + \ldots + 2^{n_s-1}) = n_s (s + 1) - 2^{n_s} + 1,
\end{align*}
\]

where, again, we have taken advantage of the 'telescoping' trick used in deriving (136); so that, by (165),

\[ T_s = (s + 1)M - n_s s + 2^{n_s} - \phi - 2. \quad (166) \]

Note that the \( T_s \), and, therefore, by (160), also \( A_s \) and \( S_s \), are all independent of the \( b \)-value \( b^* \). Furthermore, by (159), the nodes \( N_{2^v} \) and \( N_{2^v+1} \) [right-children of the children of the parent of \( N_v \)] will share the values of \( T_s' = T_{s^*}, A_s' = A_{s^*}, \) and \( S_s' = S_{s^*} \); and, once \( v \geq 2^{M-\phi-1} \) [i.e., by the argument between (158) and (159), once \( N_{2^v} \) and \( N_{2^v+1} \) are out of the apex],

\[ T_s' = T_{s^*} = (2s + Y + 1)M - (n_s + 1) (2s + Y) + 2^{n_s+1} - \phi - 2 \]

\[ = 2T_s - 2s + Y (M - n_s - 1) - (M - \phi - 2). \quad (167) \]

Until then, \( n_s \) and \( s \) both remain zero.

The total number of occurrences of \( b^* \) in Levels 0 through \( k = 2M - \phi - 2 \) is, by Lemma 20, \( 2^{k-M+\phi} = 2^{M-2} \), and so \( s \) will run from 0 through \( 2^{M-2} - 1 \). Even though, as we have seen, we can economize by using the same parameters for all values of \( b^* \) [see (160)], it is still not practical to store such a large number of coefficients (typically, as we have noted, \( M = 48 \) and \( 2^{M-2} = 7 \times 10^{13} \)), so they must be
computable when needed. To do this, we return to the concept of a node record, carrying all the information needed to generate both the left-slanting 'regular' branch or sequence, and any right-children, whenever the latter are required. In order to generate the regular left-slanting branch according to the generator $\Phi_\mu = \alpha(a, b_\mu, x_{\mu0})$, it suffices that the node-record should carry $b_\mu$ and $x_{\mu m}$, where, as in (88), $v = 2^m(2\mu + 1)$. The record $R_{2v} = (2^{m+1}N_\mu, b_\mu, x_{\mu(m+1)})$ can be obtained from $R_v = (2^mN_\mu, b_\mu, x_{\mu m})$ [see (89)] by (129) and (130), as in Algorithm 1. However, this record will have to be extended, to carry all the information needed to generate any right-children that may be needed; and we shall denote this expanded record by

$$R_v^* = [R_v; C_v], \quad (168)$$

where $C_v$ denotes the additional information. By Definition 7, like $a$ and $M$ (or $Q = 2^M$); $\phi$ (or $2^\phi$), $\psi$ (or $2^\psi$), $b_0 = 2\theta + 1$, and $f_0$ are universal parameters of the algorithm. The record $R_v$ thus suffices also to enable us to compute the node-number, $N_v = 2v + 1$, of the right-child and the new $b$-value, $b_v = b^* = <2^\phi v + b_0|Q> \quad [\text{see (134)}]$. However, to determine $x_{\nu0}$ by (160) and (161), with a possible parity-skip, we need, apart from the universal parameters $\psi$ and $f_0$, and their derived parameter $x^*$, to have the coefficients $A_s$ and $S_s$.

We note, by Lemma 13 with (94), that $a^{2M-2} \equiv 1 \pmod{2^M}$; whence

$$a = a^{2M-2-1} \quad (169)$$

acts as the reciprocal of $a$, modulo $2^M = Q$; in the sense that a factor $a^r$, appearing in any integer-valued product, reduced modulo $Q$, may be replaced by the factor $a^r$. [Suppose that such a product is $X = Ya^r$; where $X$ must be an integer, by our hypothesis. Then $Y = Xa^{-r}$; and, therefore, $Ya^r = Xa^r a^{-r} = X(a^r)^{-1} = X \equiv X = Y a^{-r} \pmod{Q}$.]. Thus, by (160), (166), and (169),

$$A_s = <a^{T_s}|Q> = <a^{(s+1)M-n_s+2^m-\phi-2}|Q> \quad (170)$$

When we turn to the other coefficient, $S_s = S_{T_s}(a)$, that we shall need to carry at every node, we first need to establish some straightforward properties of the function $S_n(z)$. 
Lemma 24. For any non-negative integers \( p \) and \( q \), and real \( z \),
\[
S_{p+q}(z) = S_p(z) + z^p S_q(z); \tag{171}
\]
\[
S_{p-q}(z) = S_p(z) - z^{p-q} S_q(z), \quad \text{if } p \geq q; \tag{172}
\]
\[
S_{pq}(z) = S_p(z) S_q(z^p); \tag{173}
\]
and, in particular,
\[
S_{2p}(z) = (1 + z) S_p(z^2) = (1 + z^p) S_p(z). \tag{174}
\]

<<We refer to the definitions in equation (5). If \( z = 1 \), then, by (6), \((\forall n \geq 0) z^n = 1 \) and \( S_n(1) = n \); whence (171) - (174) all hold, as is trivial to verify. Similarly, if \( p = 0 \) or \( q = 0 \), or \( p = q \) in (172), then (171) - (174) all hold, trivially. Suppose, therefore, that \( z \neq 1 \), \( p > 0 \), \( q > 0 \), and \( p > q \) in (172). Then, first,
\[
S_{p+q}(z) = 1 + z + z^2 + \ldots + z^{p-1} + z^p + z^{p+1} + \ldots + z^{p+q-1}
\]
\[
= (1 + z + z^2 + \ldots + z^{p-1}) + z^p (1 + z + z^2 + \ldots + z^{q-1}),
\]
from which (171) follows at once. Replacing \( p \) by \( p' \) in (171) and rearranging terms, we get
\[
S_p(z) = S_{p+q}(z) - z^p S_q(z); \tag{175}
\]
whence (172) follows immediately, when we write \( p' = p - q \). Now, by repeated application of (171), we see that
\[
S_{pq}(z) = S_{p+p'(q-1)}(z) = S_p(z) + z^{p} S_{p'(q-1)}(z)
\]
\[
= S_p(z) + z^{p} S_p(z) + z^{2p} S_{p'(q-2)}(z)
\]
\[
= S_p(z) + z^{p} S_p(z) + z^{2p} S_p(z) + z^{3p} S_{p'(q-3)}(z)
\]
\[
= \ldots = S_p(z) [1 + z^p + z^{2p} + z^{3p} + \ldots + z^{(q-1)p}],
\]
which yields (173). Finally, we note that the equality of the first and second members of (174) is Lemma 1 [equation (8)], while, if we put \( q = p \) in (171), we get the equality of the first and third members of (174). Also, the same two identities are obtained, respectively, by putting \( p = 2 \) (and then replacing \( q \) by \( p \)) and by putting \( q = 2 \), in (173).>>

By (166), (169), and (171) - (173) of Lemma 24, we see that
\[
S_{T_s}(a) = S_{(s+1) \text{M}-n_s+2r_s-\phi-2}(a) = S_{\text{M}(s-n_s+2r_s+M-\phi-2}(a)
= S_{\text{M}(s-n_s+2r_s}(a) + a^{M-s-n_s+2r_s} S_{M-\phi-2}(a)
= S_{\text{M}(s-n_s}(a) + a^{M-s-n_s} S_{n_s}\{S_{2r_s}(a) + a^{2r_s} S_{M-\phi-2}(a)\}
= S_{\text{M}(a) - a^{M-s-n_s} \{S_{n_s}\{S_{2r_s}(a) + a^{2r_s} S_{M-\phi-2}(a)\}\}
= S_{n_s}(a) S_{\text{M}(a)} - a^{M-s-n_s} \{S_{n_s}\{S_{2r_s}(a) + a^{2r_s} S_{M-\phi-2}(a)\}\};
\]
so that, by (160) and (176),
\[
S_s = <S_{T_s}(a)|\mathcal{Q}>
= \langle S_M(a) S_s(a^M) - a^{M-s-n_s} \{S_{n_s}(a) - a^{2r_s} S_{M-\phi-2}(a)\}|\mathcal{Q}\rangle.
\]

An examination of (170) and (177) reveals the parameters which need to be carried in the record \(\mathcal{R}_v^\bullet\) and updated from father-node to children, to execute the algorithm. [The need for some of these will only be seen when the details of the algorithm are examined.] The supplementary universal parameters of the algorithm (i.e., those independent of \(s\)),
\[
K_0 = <S_M(a)|\mathcal{Q}>, \quad K_0^* = <S_{M-\phi-2}(a)|\mathcal{Q}>,
K_1 = <a^M|\mathcal{Q}>, \quad K_1^* = <a^{M-\phi-2}|\mathcal{Q}>,
K_2 = <a^2|\mathcal{Q}>, \quad K_2^* = <a^{2M-2}|\mathcal{Q} > = <\bar{a}^2|\mathcal{Q}>,
\]
are computed once and for all, and stored with \(M\) (and \(Q = 2^M\)), \(a, \phi, \psi, b_0, f_0\), and \(\bar{a}\), to be used at all nodes. [The congruence for \(K_2^*\) is a consequence of Lemma 13 with (94).] This leaves thirteen variable (\(s\)-dependent) coefficients to be added to \(\mathcal{R}_v\) to make up \(\mathcal{R}_v^\bullet\); namely,
Using (159) and Lemma 24, we can now compute the update-relations for these. As was remarked at (148) and (149), and by (154), when \( 0 \leq v < 2^{M-\phi} \), \( n_s = s = 0 \); whence, for all such \( v \),

\[
\begin{align*}
U_s &= V_s = W_s = U_s^* = V_s^* = W_s^* = X_s^* = X_s^+ = 1, \\
Y_s &= Z_s = Y_s^* = Z_s^* = 0, \quad \text{and} \quad X_s = a.
\end{align*}
\]

(180)

Now, note that the bit \( Y = B_{M-\phi} \) can only be 0 or 1, and \( S_0(z) = 0 \) and \( S_1(z) = 1 \); so that

\[ S_Y(z) = Y \quad \text{and} \quad S_{1Y}(z) = Y S_0(z). \]

(181)

Similarly (although it is usually simplest to take \( z^Y \) as the conditional: \( z^Y = 1 \) if \( Y = 0 \), \( z^Y = z \) if \( Y = 1 \)), it can be useful, instead, to use the identity

\[ z^Y = 1 + (z - 1)Y. \]

(182)

Also, repeated factors \( Y \), in the same term can be simplified, since

\[ Y^2 = Y \quad \text{and} \quad Yz^Y = Yz. \]

(183)

Thus, for all \( v \geq 2^{M-\phi-1} \) [when either \( s = 0 \) and \( Y = 1 \), or \( s \geq 1 \); as is noted between (158) and (159)], we have, in the simplest terms, modulo \( Q \):

\[
\begin{align*}
U_{s'} &= U_{s''} = a^{2(2s+Y)} = U_s^2 K_2^Y, \\
V_{s'} &= V_{s''} = a^{n_{s+1}} = a^n a = V_s a.
\end{align*}
\]

(184)

(185)
\[ W_s' = W_{s'} = a^{n_s+1}(2s+Y) = a^{2n_s+2s+n_sY+Y} \]
\[ = W_s^2 U_s (V_s a)^Y = W_s^2 U_s V_s^Y. \tag{186} \]
\[ U_s^* = U_{s^*} = a^{2(2s+Y)} = U_s^* 2 K_2^Y, \tag{187} \]
\[ V_s^* = V_{s^*} = a^{n_s+1} = a^{n_s} a = V_s^* a, \tag{188} \]
\[ W_s^* = W_{s^*} = a^{(n_s+1)(2s+Y)} = a^{2n_s+2s+nsY+Y} \]
\[ = W_{s^2} U_{s^*} (V_{s^*} a)^Y = W_{s^*} U_s^* V_s^* Y. \tag{189} \]
\[ X_s' = X_{s^*} = a^{2n_s+1} = (a^{2n_s})^2 = X_s^2, \tag{190} \]
\[ Y_s' = Y_{s^*} = S_{2(2s+Y)}(a) = (1 + a^{2s+Y}) S_{2s+Y}(a) \]
\[ = (1 + U_s a^Y) [S_{2s}(a) + a^{2s} S_Y(a)] \]
\[ = (1 + U_s a^Y) (Y_s + U_s V_s Y). \tag{191} \]
\[ Z_s' = Z_{s^*} = S_{n_s+1}(a) = S_{n_s}(a) + a^{n_s} S_1(a) = Z_s + V_s, \tag{192} \]
\[ X_s^* = X_{s^*} = a^{M(2s+Y)} = X_s^* 2 K_1^Y, \tag{193} \]
\[ Y_s^* = Y_{s^*} = S_{2s+Y}(a^M) = S_{2s}(a^M) + a^{2Ms} S_Y(a^M) \]
\[ = (1 + a^{Ms}) S_{s}(a^M) + X_s^* 2 Y \]
\[ = (1 + X_s^*) Y_s^* + X_s^* 2 Y. \tag{194} \]
\[ Z_s^* = Z_{s^*} = S_{(n_s+1)(2s+Y)}(a) = S_{2n_s+2s+n_sY+Y}(a) \]
\[ = S_{2n_s+2s+2s+nsY} = a^{n_s+1} S_Y(a) \]
\[ = S_{2n_s+2s+2s+n_sY} + W_s^2 U_s V_s Y \]
\[ = S_{2n_s+2s+2s}(a) + a^{2n_s+2s} S_{n_s+1} + W_s^2 U_s V_s Y \]
\[ = S_{2n_s+2s+2s}(a) + a^{2n_s} S_{2s}(a) + W_s^2 U_s Y (Z_s + V_s) \]
\[ = (1 + a^{n_s}) S_{n_s+2s}(a) + W_s^2 [Y_s + U_s Y (Z_s + V_s)] \]
\[ = (1 + W_s) Z_s^* + W_s^2 [Y_s + U_s Y (Z_s + V_s)]. \tag{195} \]
\[ X_s^t = X_{s^t} = S_{2n_s+1}(a) = S_{2s+2n_s}(a) = (1 + X_s) X_s^t. \tag{196} \]
In terms of these coefficients, we see that (170) and (177) become:

\[ A_s = \langle K_1^* X_s^* W_s^* X_s | \mathcal{Q} \rangle \]  

(197)

and

\[ S_s = \langle K_0 Y_s^* - X_s^* W_s^* (Z_s^* - X_s^* - X_s K_0^*) | \mathcal{Q} \rangle. \]

(198)

Algorithm 2. The procedure carries at each node, numbered \( \nu = 2^m N_\mu = 2^m(2\mu + 1) \), a record [see (89), (168), and (179)]

\[ \mathbf{R}_\nu^* = [R_\nu; C_\nu] \]

\[ = [2^m N_\mu, b_\mu, x_{\mu m}; U_s, V_s, W_s, U_s^*, V_s^*, W_s^*, X_s, Y_s, Z_s, X_s^*, Y_s^*, Z_s^*, X_s^*]. \]

(199)

the transformation for which, on passage to the two child-nodes is given by

\[ \mathbf{R}_{2\nu}^* = L_{T, \phi, \psi}(\mathbf{R}_\nu^*), \quad \mathbf{R}_{2\nu+1}^* = \mathbf{R}_{N_\nu}^* = M_{T, \phi, \psi}(\mathbf{R}_\nu^*). \]

(200)

These mappings are defined as follows.

(a) for \( R_\nu \):

\[ L_{T, \phi, \psi}(\nu = 2^m N_\mu) = (2\nu = 2^{m+1} N_\mu), \]

\[ L_{T, \phi, \psi}(b\text{-value at } \nu = b_\mu) = b_\mu, \]

\[ L_{T, \phi, \psi}(x\text{-value at } \nu = x_{\mu m}) = (x_{\mu (m+1)} = \langle ax_{\mu m} + b_\mu | \mathcal{Q} \rangle); \]

and

\[ M_{T, \phi, \psi}(\nu = 2^m N_\mu) = (N_\nu = 2\nu + 1 = 2^{m+1} N_\mu + 1), \]

\[ M_{T, \phi, \psi}(b\text{-value at } \nu = b_\mu) = \langle 2^\phi \nu + b_0 | \mathcal{Q} \rangle, \]

\[ M_{T, \phi, \psi}(x\text{-value at } \nu = x_{\mu m}) \]

\[ = \left\{ \begin{array}{l} x_{\nu 0} = \langle A_s x_s^* + S_s b_s^* | \mathcal{Q} \rangle \text{ whichever has the same parity as } x_{\mu m} \end{array} \right\}; \]
where $x^* = <2^v v_0 + f_0 \mid \Omega>$, $b^* = <2^\phi v + b_0 \mid \Omega>$, and $A_s$ and $S_s$ are computed from (197) and (198), using the coefficients in $C_v$.

(b) for $C_v$: the coefficients remain at the values in (180), so long as $v < 2^{M-\phi}$; and, whenever $v \geq 2^{M-\phi-1}$,

$$L_{T, \phi, \psi}(U_s) = U_s = \mathcal{M}_{T, \phi, \psi}(U_s) \quad U_s^* = <U_s^2 K_2^\psi \mid \Omega>.$$

$$L_{T, \phi, \psi}(V_s) = V_s = \mathcal{M}_{T, \phi, \psi}(V_s) \quad V_s^* = <V_s^a \mid \Omega>.$$

$$L_{T, \phi, \psi}(W_s) = W_s = \mathcal{M}_{T, \phi, \psi}(W_s) \quad W_s^* = <W_s^2 U_s V_s \cdot \mid \Omega>.$$

$$L_{T, \phi, \psi}(U_s^*) = U_s^* = \mathcal{M}_{T, \phi, \psi}(U_s^*) \quad U_s^{**} = <U_s^{*2} K_2^\psi \mid \Omega>.$$

$$L_{T, \phi, \psi}(V_s^*) = V_s^* = \mathcal{M}_{T, \phi, \psi}(V_s^*) \quad V_s^{**} = <V_s^* a \mid \Omega>.$$

$$L_{T, \phi, \psi}(W_s^*) = W_s^* = \mathcal{M}_{T, \phi, \psi}(W_s^*) \quad W_s^{**} = <W_s^{*2} U_s^* V_s^* \mid \Omega>.$$

$$L_{T, \phi, \psi}(X_s) = X_s = \mathcal{M}_{T, \phi, \psi}(X_s) \quad X_s^* = <X_s^2 \mid \Omega>.$$

$$L_{T, \phi, \psi}(Y_s) = Y_s = \mathcal{M}_{T, \phi, \psi}(Y_s) \quad Y_s'' = <(1 + U_s a^Y) (Y_s + U_s Y) \mid \Omega>.$$

$$L_{T, \phi, \psi}(Z_s) = Z_s = \mathcal{M}_{T, \phi, \psi}(Z_s) \quad Z_s'' = <Z_s + V_s \mid \Omega>.$$

$$L_{T, \phi, \psi}(X_s^*) = X_s^* = \mathcal{M}_{T, \phi, \psi}(X_s^*) \quad X_s^{**} = <X_s^{*2} K_1^\psi \mid \Omega>.$$

$$L_{T, \phi, \psi}(Y_s^*) = Y_s^* = \mathcal{M}_{T, \phi, \psi}(Y_s^*) \quad Y_s^{**} = <(1 + X_s^*) Y_s^* + X_s^{*2} Y \mid \Omega>.$$

$$L_{T, \phi, \psi}(Z_s^*) = Z_s^* = \mathcal{M}_{T, \phi, \psi}(Z_s^*) \quad Z_s^{**} = <(1 + W_s) Z_s^* + W_s^{*2} [Y_s + U_s Y (Z_s + V_s)] \mid \Omega>.$$

$$L_{T, \phi, \psi}(X_s^+) = X_s^+ = \mathcal{M}_{T, \phi, \psi}(X_s^+) \quad X_s^{*+} = <(1 + X_s) X_s^+>.$$

where the various symbols are defined in (178) and (179).
A multiplication-count (there are no divisions, and we may suppose that the reductions modulo \( Q \) are performed by truncation of binary computer-words; also, we do not count multiplications by powers of 2, which can be performed by fast bit-shifts) yields 1 for generating the three components of \( R_{2^v} \) by \( L_{\mathcal{T}, \phi, \psi} \), and 9 or 10 for generating the three components of \( R_{2^{v+1}} \) by \( M_{\mathcal{T}, \phi, \psi} \) (including computing the current \( A_s \) and \( S_s \)); while, for generating the thirteen components of \( C_{2^v} \) and \( C_{2^{v+1}} \) (which are identical), we require 15 multiplications when \( Y = 0 \), and 7 more when \( Y = 1 \). Thus, from Level \( M - \phi - 1 \) on, the algorithm takes, altogether, an average of \( 1 + 19/2 + 15 + 7/2 = 29 \) multiplications to generate the records \( R^* \) of both children of any given node (considerably less in the apex of the tree); or an average of 14.5 multiplications per node. Since these nodes occur only every \( T \) pseudo-random numbers (as we explained earlier; see Lemma 18), and we expect \( T \) to be of the order of 10, with other steps taking 1 multiplication to perform, by (4); the overall expected number of multiplications per pseudo-random number generated will only be about 2.35; that is, between 2 and 3 times the time required by the highly-efficient linear congruential generator itself, without any tree-structure. This would appear to be very satisfactory.

6. COMPUTATIONAL RESULTS

A program was written in "C" to execute Algorithm 1. The section which inputs and computes the universal parameters of the algorithm.

<table>
<thead>
<tr>
<th>ANALYSIS</th>
<th>( M )</th>
<th>( a )</th>
<th>( b_0 )</th>
<th>( f_0 )</th>
<th>( \phi )</th>
<th>( \psi )</th>
<th>( Q = 2^M )</th>
<th>( 2^\phi )</th>
<th>( 2^\psi )</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROGRAM</td>
<td>( M )</td>
<td>( a )</td>
<td>( b_0 )</td>
<td>( f_0 )</td>
<td>( q_0 )</td>
<td>( q_x )</td>
<td>( Q )</td>
<td>( q_0 q_0 )</td>
<td>( q_x q_x )</td>
</tr>
</tbody>
</table>

and initializes the first record, at the root of the tree, takes just 13 commands ["scanf("")" being taken as one command, and "for (...)" being counted as a command additional to what it controls]:

```c
scanf("%ld %ld %ld %ld %ld", &M, &a, &b0, &f0, &q0, &q_x); Q = 1; for (i = 0; i < M; i++) Q = Q * 2;
q_0 = 1; for (i = 0; i < q_0; i++) q_0 = q_0 * 2;
q_x = 1; for (i = 0; i < q_x; i++) q_x = q_x * 2;
j = 1; bval[j] = b0; xval[j] = f0;
```
Here, "bval[j]" stores the b-value and "xval[j]" stores the x-value, at node number "j".

The section which "builds the tree", i.e., computes the records at the child-nodes of a given parent-node, takes 7 commands, including the "for ( . . . )" loop over all parent nodes:

for the left-child of node "i":

```
++j;
bval[j] = bval[i];
xval[j] = res(a * xval[i] + bval[i]);
```

for the right-child of node "i":

```
++j;
bval[j] = res(qq * i + b0);
xval[j] = xval[i];
```

Here, "res(x)" denotes \( \langle x \rangle \), the residue of "x" modulo "q".

With \( M = 6, \phi = 3, \) and \( \psi = 4 \), the computations covered the first 255 nodes of the tree (8 levels). The following eight sets of data were taken.

\[
\begin{array}{cccccccc}
a & 21 & 37 & 5 & 53 & 45 & 13 & 21 & 5 \\
b_0 & 3 & 63 & 7 & 11 & 33 & 11 & 33 \\
f_0 & 7 & 57 & 5 & 1 & 37 & 33 & 0 & 42 \\
\end{array}
\]

For every set of data, identical patterns of numbers of repetitions of initial \((b, x)\)-values were observed:

- Level 0: 0 repetitions
- Level 1: 0 repetitions
- Level 2: 0 repetitions
- Level 3: 0 repetitions
- Level 4: 3 repetitions
- Level 5: 7 repetitions
- Level 6: 16 repetitions
- Level 7: 35 repetitions
- Total: 61 repetitions

Since the values of \( a \) [subject only to (95)], \( b_0 \) [odd], and \( f_0 \) were chosen quite artlessly, the recurring pattern of repetitions suggests that a theorem underlies it: the number of repetitions at each level is probably a constant, depending only on \( M, \phi, \) and \( \psi \). Further experimentation, varying \( M \) and \( \phi \), supports this conjecture. For example, covering the first 511 nodes, when \( M = 7, \phi = 5, \) and \( \psi = 6, \)
we get; both for \( a = 5, \ b_0 = 5, \) and \( f_0 = 5, \) and for \( a = 37, \ b_0 = 23, \) and \( f_0 = 30; \) that the following patterns of numbers of repetitions of initial \((b, x)\)-values occurred:

- Level 0: 0 repetitions
- Level 1: 0 repetitions
- Level 2: 0 repetitions
- Level 3: 2 repetitions
- Level 4: 4 repetitions
- Level 5: 9 repetitions
- Level 6: 17 repetitions
- Level 7: 22 repetitions
- Level 8: 21 repetitions
- Total: 75 repetitions

Another program was written in "C" to execute Algorithm 2. The section which inputs and computes the universal parameters of the algorithm and their immediate derivates,

<table>
<thead>
<tr>
<th>ANALYSIS</th>
<th>( M )</th>
<th>( a )</th>
<th>( b_0 )</th>
<th>( f_0 )</th>
<th>( \phi )</th>
<th>( \psi )</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROGRAM</td>
<td>( M )</td>
<td>( a )</td>
<td>( b_0 )</td>
<td>( f_0 )</td>
<td>( \phi )</td>
<td>( \psi )</td>
</tr>
</tbody>
</table>

| ANALYSIS | \( 2^\phi \) | \( 2^\psi \) | \( 2^{M-\phi-1} \) | \( 2^{M-\phi} \) | \( 2^{M-\phi+1} \) | \( Q = 2^M \) |
| PROGRAM   | \( q\#b \) | \( q\#x \) | \( Q \) | \( Q \) | \( Q \) | \( Q \) |

now takes 13 commands:

```c
scanf("%ld %ld %ld %ld %ld", &M, &a, &b0, &f0, &qb, &qx); 
Q0 = Q1 = Q2 = 1; for (i = 0; i < QB; i++) q\#b = q\#b * 2; 
Q = q\#b * Q0; 
QX = 1; for (i = 0; i < qx; i++) q\#x = q\#x * 2; 
```

That which computes the parameters in (169) and (178).

<table>
<thead>
<tr>
<th>ANALYSIS</th>
<th>( K_0 )</th>
<th>( K_0^* )</th>
<th>( K_1 )</th>
<th>( K_1^* )</th>
<th>( K_2 )</th>
<th>( K_2^* )</th>
<th>( \dot{a} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROGRAM</td>
<td>( K0 )</td>
<td>( KK0 )</td>
<td>( K1 )</td>
<td>( KK1 )</td>
<td>( K2 )</td>
<td>( KK2 )</td>
<td>( \ddot{a} )</td>
</tr>
</tbody>
</table>

takes 20 commands:

```c
KK0 = 0; u = M - q - 2; v = 1; 
for (i = 0; i < u; i++) 
{ KK0 = res(KK0 + v); 
  v = res(v * a); 
}
KK1 = v; K0 = KK0;
```
for (i = u; i < M; i++)
    { K0 = res(K0 + v);
       v = res(v * a);
    }
K1 = v; K2 = res(a * a);
aa = a; u = a;
for (i = 3; i < M; i++)
    { u = res(u * u);
       aa = res(aa * u);
    }
KK2 = aa; KK2 = res(KK2 * KK2);

The initialization of the first record, at the root of the tree, takes 3 commands, as before:

\[ j = 1; bval[j] = b0; xval[j] = f0; \]

There remain the fifteen special coefficients [see (179) and (180)].

<table>
<thead>
<tr>
<th>ANALYSIS</th>
<th>S</th>
<th>(n_s)</th>
<th>U_s</th>
<th>V_s</th>
<th>W_s</th>
<th>U_s^*</th>
<th>V_s^*</th>
<th>W_s^*</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>ANALYSIS</th>
<th>X_s</th>
<th>Y_s</th>
<th>Z_s</th>
<th>X_s^*</th>
<th>Y_s^*</th>
<th>Z_s^*</th>
<th>X_s^†</th>
</tr>
</thead>
</table>

These are the same throughout the apex of the tree; but, because the apex is, for efficiency, "built" more simply than the rest of the tree, we do not need them in the body of the apex. We must, however, initialize the coefficients in Level \(M - \phi\), and this takes 16 commands, including the "for (...)" loop over all nodes in this level:

\[
\begin{align*}
st[i] &= 0; nt[i] = 0; XXX[i] = 1; \\
U[i] &= 1; V[i] = 1; W[i] = 1; \\
UU[i] &= 1; VV[i] = 1; WW[i] = 1; \\
X[i] &= a; Y[i] = 0; Z[i] = 0; \\
XX[i] &= 1; YY[i] = 0; ZZ[i] = 0;
\end{align*}
\]

The section which "builds the apex" takes 11 commands, including the "for (...)" loop over all nodes in the apex ["if (...)...else..." being taken as a command, additional to what it controls]:

---

- 67 -
Finally, the section which builds the rest of the tree takes 63 commands, including the "for(...)" loop over all nodes, and the copying of all coefficient values (which are common to both left and right children of any given node):

for the left-child of node "i":

```
for the left-child of node "i":
```

```
for the left-child of node "i":
```

```
for the right-child of node "i": first, copy all fifteen coefficients; then:
```
Here, "A[i]" denotes $A_s$ at node "i" and "s[i]" denotes $S_s$ at node "i", respectively computed per (197) and (198).

Thus, the avoidance of repetitions in the first $k + 1 = 2M - \phi - 1$ levels of the tree [and commensurate avoidance of repetitions thereafter, within $2^{k+\phi} = 2^{M-2}$ occurrences of any $b$-value] costs a factor of $127/20 = 6.35$ in program-complexity. Note that what we have counted above are commands in the program listing, not executions (which are counted at the end of §5, for a factor of only 2.325 in computation time).

With $M = 6$, $\phi = 3$, and $\psi = 4$, the computations again covered the first 255 nodes (8 levels) of the tree. The same eight sets of data were tried, yielding the theoretically predicted absence of repetitions in the first $k + 1 = 2M - \phi - 1 = 8$ levels. This confirms the efficacy of Algorithm 2.

7. CONCLUSIONS

We have presented here, in full and rigorous detail, the theory governing the linear congruential type of pseudo-random generator, as defined in (1) - (4). In keeping with the most frequent practice, we have concentrated on the length of the period of such sequences. Linear congruential sequences are periodic [Lemma 7] and have no repetition of $x$-values in a period [Lemma 17]. Under the conditions that $a = 1 \pmod{4}$ and $b = 1 \pmod{2}$, the sequences are completely periodic [Lemma 9], with period $Q = 2^M$ [Lemma 12]. This means that the sequence $[x_j]_{j=0}^{\infty}$ [defined in (2) - (4)] will, in every period, pass just once through each integer value in the interval $[0, Q - 1]$. By (1), the (similarly periodic) rational sequence $[\xi_j]_{j=0}^{\infty}$ will thus pass exactly once in every period through each integer multiple of $2^{-M}$ in the real interval $[0, 1)$; yielding pseudo-random numbers whose distribution in
[0, 1) is quite close to canonical uniformity. This property, given qualitative ("uniformity") and quantitative ("coarseness") precision in Definition 4, is, of course, crucial to the usefulness of such a sequence in performing Monte Carlo computations.

Turning to branching processes, such as are useful in many Monte Carlo computations, we seek to define families of linear congruential generators which are easy to specify at any node of a binary tree, without storing all possible sets of parameters, since the growth of such storage would rapidly become completely prohibitive. Seeking a criterion which is both tractable and useful, for the independent behavior of sequences \( [x^i] \) and \( [x^+ j] \) we look at the difference sequence \( [\delta j] \) (defined in (26)) and go, by analogy with the concepts of uniformity and coarseness, to those of "independence" and "consonance" given in Definition 5. A rather thorough analysis of this criterion is given here, giving conditions for low consonance between sequences generated at nodes which are close to each other in the tree. Further analysis, of discrepancies [see HAL 60, HAL 70, HAL 72, HAM 60, HAM 64, NIE 78, ZAR 66, and ZAR 68] and correlations of such proximate sequences, is envisaged for future research, to reinforce the results presented here.

Three algorithms for the generation of suitable families of linear congruential sequences are analyzed here. The first is due to Warnock [WAR 83] and the other two (herein named Algorithms 1 and 2) are new. Algorithm 1 is similarly simple to Warnock's, but has (as does Warnock's algorithm) some problems, in this case related to the rather early occurrence of repeated generators. These problems are addressed and substantially alleviated in Algorithm 2. It is calculated that the code required for the second, improved algorithm is six or seven times longer than for the first; and that the computation time required per random number is two to three times longer than is required by the basic (highly efficient) linear-congruential generator.

While more research can usefully be done on this new tool, it would appear that a powerful and efficient technique has been introduced here, with considerable theoretical support.
ACKNOWLEDGEMENTS

I am grateful to several colleagues, and especially to Dr T. T. Warnock of Cray Research, Inc., for some interesting and stimulating discussions of the applications, desirable properties, and necessary limitations of tree-structured random generators of the type developed here. I am also very much indebted to Dr T. E. Booth of Los Alamos National Laboratory for an exceptionally careful reading of earlier versions of the manuscript, and for many relevant comments.
BIBLIOGRAPHY


<table>
<thead>
<tr>
<th>Reference</th>
<th>Title</th>
<th>Author(s)</th>
<th>Source</th>
</tr>
</thead>
</table>