Theoretical Computer Science 44 (1986) 209-224
North-Holland

AN ALGORITHM FOR GENERATING SHIFT-REGISTER CYCLES

T. ETZION

Computer Science Department, Technion-Israel Institute of Technology, Haifa 32000, Israel

Communicated by M. Nivat
Received March 1985

Abstract. Given two integers n and L, an algorithm to construct binary sequences of length L such that $1 \leq L \leq 2^n$ and no n-tuple appears twice in a sequence, is given. This algorithm can be generalized to m-ary sequences, i.e., to construct m-ary sequences of length L such that $1 \leq L \leq m^n$ and no n-tuple appears twice in a sequence.

1. Introduction

In several applications of digital communication systems it is desired to construct an m-ary sequence of length L such that $1 \leq L \leq m^n$ and each n-tuple appears at most once in the sequence. Lempel [7] proved the existence of such a sequence for every m, n, and L. Hemmati and Costello [6] gave an algebraic algorithm to construct m-ary sequences of arbitrary length L, $1 \leq L \leq m^n$, where m is a power of a prime number. In this paper we give an algorithm to produce binary sequences for any given length L, $1 \leq L \leq 2^n$, such that each n-tuple appears at most once in the sequence. This algorithm can be generalized to m-ary sequences, i.e., to construct m-ary sequences of length L such that $1 \leq L \leq m^n$ and no n-tuple appears twice in a sequence.

2. Construction of L-cycles

In this section we describe a construction of binary shift-register cycles of any given length L. The proposed construction proceeds along the following lines.

Given the desired cycle length L, let n be the integer such that $2^{n-1} < L \leq 2^n$. We join cycles from the pure cycling register of order n (PCR_n), in appropriate order, to obtain a cycle of length L', where $L \leq L' < L+n$. Expressing the difference $L'-L$ in base-2 notation:

$$L'-L=\sum_{i=0}^{\lfloor \log(n-1) \rfloor} l_i 2^i,$$

we form an L-cycle by removing from the L'-cycle one cycle of length 2^i for each i such that $l_i = 1$. It can be easily verified that if each n-tuple appears at most once in the cycle, also each k-tuple, $k \geq n$, appears at most once in the cycle.

Before we can go into more specific details of the proposed construction, we need some preparation in way of definitions and notation.

Consider the set B^n of all binary n-tuples forming the set of all states of a binary feedback shift-register of length n. The *companion* X' of a state $X = (x_1, x_2, \ldots, x_n) \in B^n$ is defined by

$$X' = (x_1, x_2, \ldots, x_{n-1}, \bar{x}_n),$$

where $\bar{x}$ denotes the complement of x.

For $X, Y \in B^n$ the relation $X \to Y$ is defined by

$$X \to Y \text{ iff } (x_2, \ldots, x_n) = (y_1, \ldots, y_{n-1}).$$

A *k-cycle* C (a cycle of length k) is a cyclic sequence of k distinct states, $C = [X^{(1)}, X^{(2)}, \ldots, X^{(k)}]$, such that $X^{(k)} \to X^{(1)}$ and $X^{(i)} \to X^{(i+1)}$, $i = 1, 2, \ldots, k-1$. If $X^{(i)} = (x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)})$, then C can also be represented by the cyclic binary sequence $C = [X_1^{(1)}, X_1^{(2)}, \ldots, X_1^{(k)}]$, where (cyclically) contiguous n-tuples correspond to states.

Two cycles C_1 and C_2 are said to be *adjacent* if they are (state) disjoint and there exists a state X on C_1 such that its companion X' is on C_2. A cycle C is *reducible* if there exists a state X on C such that X' is also on C.

Theorem 2.1 ([4, p. 51]). *A reducible cycle C is split into two adjacent cycles when the predecessors of X and X' are interchanged. Two adjacent cycles C_1 and C_2 with X on C_1 and X' on C_2 are joined into a single cycle when the predecessors of X and X' are interchanged.*

The PCR_n is a feedback shift-register whose cycles have the following property: $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2, \ldots, y_n)$ are on the same PCR_n-cycle iff X is a cyclic shift of Y. It is well known [4] that the length of a cycle from the PCR_n is a divisor of n. The PCR_n was also used in [2, 5] to produce De Bruijn cycles.

The *weight* $W(X)$ *of a state* X is the number of ones in $X = (x_1, x_2, \ldots, x_n)$, i.e., $W(X) = \sum_{i=1}^{n} x_i$. Clearly, states belonging to the same cycle of the PCR_n have the same weight. The *weight* $W(C)$ *of a cycle* C from the PCR_n is the weight of each of its states.

For further reference, we present the following obvious observation in the form of a lemma.

Lemma 2.2. *Let C_1 be a cycle of weight $k > 0$ from the PCR_n. Then there exists a state $X = (x_1, x_2, \ldots, x_n)$ on C_1 such that its companion X' is on a cycle C_2 whose weight is $k-1$.*

2.1. The main cycle

Theorem 2.1 and Lemma 2.2 together with results derived in the sequel suggest a method of producing L-cycles via the joining of (PCR_n-)cycles. At each step of the proposed method we have a *main cycle* (*MC*), obtained in previous steps by joining a subset of the PCR_n-cycles and the remaining PCR_n-cycles. In order to simplify the description, assume that, initially, the MC is an appropriately formed cycle containing all the states of weight less than four. Generally, in the beginning of step i, $i \geq 1$, the MC contains all the states of weight less than $i+3$. The MC is then extended by joining to it all the PCR_n-cycles of weight $i+3$, with a possible exception in the last step where the required length is reached before all the PCR_n-cycles of the final weight have been used. This is always possible because the current MC contains all the states of weight $i+2$ (see Theorem 2.1 and Lemma 2.2). This part of the procedure ends when the MC reaches length L' such that $L \leq L' < L+n$.

To detect the completion of this phase of the algorithm we have to determine in advance the weight m and length h of the last PCR_n-cycle to be joined to the MC, as well as the total number t on cycles of weight m and length h that take part in forming the final MC. Since the number of states of weight i is $\binom{n}{i}$, the number of states of weight less than or equal to j is $\sum_{i=0}^{j}\binom{n}{i}$. Hence, the final weight m is determined by the inequalities

$$\sum_{j=0}^{m-1}\binom{n}{j} < L \leq \sum_{j=0}^{m}\binom{n}{j}. \tag{1}$$

Since $2^{n-1} < L \leq 2^n$, it follows that

$$m \geq \lfloor \tfrac{1}{2}(n+1) \rfloor. \tag{2}$$

To determine the final length h we first need to discuss another class of cycles used in our construction.

Let $Y_i = 0^{2^{i-1}}1$, where 0^k denotes a sequence of k zeros, and consider the cycles of the form $[Y_i]$, $1 \leq i < \log n$. The removal of selected $[Y_i]$-cycles from the final MC serves to reduce its length L' to the desired length L. If

$$L' - L = \sum_{i=0}^{\lfloor \log(n-1) \rfloor} l_i 2^i, \tag{3}$$

the nonzero l_i's, $i \geq 1$ indicate the values of i for which a $[Y_i]$-cycle has to be removed from the final MC. In addition, if $l_0 = 1$, the cycle [0] must also be removed.

To ensure that, when required, the removal of $[Y_i]$-cycles is possible, it suffices to make sure, first, that all the states belonging to such $[Y_i]$-cycles are on the final MC and, second, that all the states from the same $[Y_i]$-cycle are contiguous.

From (2), it is clear that, for $i \geq 2$, the weight of every state from a $[Y_i]$-cycle is less than m and, therefore, each such state is on the final MC. If $m > \lfloor \frac{1}{2}(n+1) \rfloor$, then the same is true for the $[Y_1]$-cycle. Thus, the only case to be provided for is the case where $i = 1$ and $m = \lfloor \frac{1}{2}(n+1) \rfloor$. In this case, $Y_1 = 01$ and we distinguish

between even and odd n. For even n, both states of $[Y_1]$ belong to the PCR_n-cycle of length 2 and weight $m=\frac{1}{2}n$ and we stipulate that this be the first cycle of weight m to join the MC. For odd n the two states of $[Y_1]$ are $((01)^{(n-1)/2}0)$ and $((10)^{(n-1)/2}1)$. Only the second state has weight $m=\frac{1}{2}(n+1)$ and we stipulate that the PCR_n-cycle $[(10)^{(n-1)/2}1]$ to which it belongs be the first cycle of weight m to join the MC (note that the length of this cycle is n). This ensures that all the states belonging to $[Y_i]$-cycles are on the final MC. The contiguity on the final MC of the states belonging to a $[Y_i]$-cycle will be discussed at a later point.

To decide the final length h, we need the following lemma.

Lemma 2.3. *Let $N(k, \varepsilon)$ be the number of PCR_k-cycles of length k and weight $W=\varepsilon k$, $0\leq\varepsilon\leq 1$. Then*

$$N(k,\varepsilon)=\frac{1}{k}\sum_{d|k}\binom{d}{\varepsilon d}\mu\left(\frac{k}{d}\right),$$

where $\mu(\ )$ is the Möbius function and the binomial coefficient $\binom{a}{b}$ is defined as zero if b is not an integer.

Proof. The total number of states of weight $W=\varepsilon k$ in the PCR_k is $\binom{k}{\varepsilon k}$. On the otherhand, the number of PCR_k-states with $W=\varepsilon k$ is also equal to $\sum_{d|k} dN(d,\varepsilon)$. Therefore, $\binom{k}{\varepsilon k}=\sum_{d|k} dN(d,\varepsilon)$ and by the Möbius Inversion Theorem [3],

$$N(k,\varepsilon)=\frac{1}{k}\sum_{d|k}\binom{d}{\varepsilon d}\mu\left(\frac{k}{d}\right). \qquad \square$$

Having joined the $[Y_i]$-cycles of weight m, if any, to the MC we proceed by joining other PCR_n-cycles of weight m, in order of decreasing length. Since the number of states of weight m belonging to the PCR_n-cycles of length greater than l is $\sum_{d|n,d>l} dN(d, m/n)$, the final length h is determined by the inequalities

$$\sum_{d|n,d>h} dN\left(d,\frac{m}{n}\right)<L-\sum_{j=0}^{m-1}\binom{n}{j}-p(m,n)\leq\sum_{d|n,d\geq h} dN\left(d,\frac{m}{n}\right), \tag{4}$$

where

$$p(m,n)=\begin{cases}2 & \text{if } m=\frac{1}{2}n,\\ 0 & \text{otherwise.}\end{cases}$$

The total number t of *h-cycles of weight m* that take part in the formation of the final MC can now be readily determined by the following inequalities

$$0<L-\sum_{j=0}^{m-1}\binom{n}{j}-p(m,n)-\sum_{d|n,d>h} dN\left(d,\frac{m}{n}\right)-h(t-1)\leq h. \tag{5}$$

Having identified the final weight m, the final length h, and the number t we can write

$$L' = \sum_{j=0}^{m-1} \binom{n}{j} + p(m, n) + \sum_{d|n, d>h} dN\left(d, \frac{m}{n}\right) + ht.$$

2.2. *Selection of bridging states*

We proceed now to describe in full detail the various steps of forming the final MC.

Consider a cycle C from the PCR_n. Let $S(C)$ be the state of C whose value $|S(C)|$, when viewed as a number in base-2 notation, is maximal. Writing $|S(C)| = k2^r$, where k is odd and $r \geq 0$, it is easy to see that the states $S^*(C)$ such that $|S^*(C)| = k$ is also on C. For, if $C \neq [0]$ we can write $S(C) = (\alpha 0^r)$, where $a \in B^{n-r}$ begins and ends in a one and, therefore, $S^*(C) = (0^r a)$.

In the process of forming the final MC, a PCR_n-cycle C is joined to the current MC through a pair of companion states X and X' belonging to C and to the current MC, respectively. The state X on C will be referred to as the *bridging state* of the join. By Theorem 2.1, interchanging the predecessors of X and X' will create the next MC by joining the current one with C.

In most cases $S^*(C)$ can be chosen as the bridging state of C. The only exceptions to this rule arise as a result of the requirement that, on the final MC, the states belonging to the same $[Y_i]$-cycle, $1 \leq i < \log n$, be contiguous. This requirement is satisfied by ensuring that, at each step of the construction, states on the current MC that belong to the same $[Y_i]$-cycle are contiguous.

To define the exceptions to the rule of choosing $S^*(C)$ as the bridging state of C, we check the given value of n against each i, $1 \leq i < \log n$. By Corollary A.8, exceptions only arise when i is such that 2^i does not divide n. (This and other results cited in the sequel are presented in Appendix A.)

When n is not divisible by 2^i, we can write

$$n - 1 = 2^i q_i + r_i, \quad q_i > 0, 0 \leq r_i \leq 2^i - 2. \tag{6}$$

In this case, $[Y_i]$ is not a PCR_n-cycle and, by the proof of Lemma A.1, its states belong to the following PCR_n-cycles:

$$C_i^1 = [0^{r_i+2^i} 1 Y_i^{q_i-1}], \qquad C_i^2 = [0^{r_i} 1 Y_i^{q_i}].$$

It also follows from the proof of Lemma A.1 that choosing $S^*(C_i^1)$ and $(0^{r_i} 1 Y_i^{q_i}) \neq S^*(C_i^2)$ as the bridging states of C_i^1 and C_i^2 respectively, preserves the contiguity of the states of this $[Y_i]$-cycle on the MC. It can also be shown that using $S^*(C_i^2)$ as the bridging state for C_i^2 will interfere with the said contiguity.

By Corollary A.6, the only other exception to the rule of choosing $S^*(C)$ arises when $i = \lfloor \log n \rfloor$ and the cycle C to be joined to the current MC is either of the form $[0^{n-k-2}10^k1]$, $n - 2^{\lfloor \log n \rfloor} \leq k < \frac{1}{2}(n-2)$ or $C = [0^{(n-2)/2}1]$. By Lemma A.9, to preserve contiguity in such a case, we can choose $(10^k 10^{n-k-2})$ as the bridging state for C instead of $S^*(C)$.

To summarize, let 2^{i_0} be the highest power of 2 that divides n. If $i_0 = \log n$, there are no exceptions to the rule of choosing $S^*(C)$. If $i_0 < \log n$, the *exception set* of PCR_n-cycles C for which the bridging state is not $S^*(C)$ consists of

(1) the cycles of the form $[0^{r_i} 1 Y_i^{q_i}]$, $i_0 < i < \log n$, and

(2) the cycles of the form $[0^{n-k-2}10^k 1]$, $n - 2^{\lfloor \log n \rfloor} \leq k < \frac{1}{2}(n-2)$ and $[0^{(n-2)/2}1]$.

2.3. *Removal of* $[Y_i]$*-cycles*

Having formed the final MC, we apply Theorem 2.1 to remove from it those $[Y_i]$-cycles for which $l_i = 1$ in (3); we also remove the cycle [0] if $l_0 = 1$. To determine the pair of companion states through which such a $[Y_i]$-cycle is to be removed, we check as before the given value of n against each i, $1 \leq i < \log n$.

When n is not divisible by 2^i, it follows from Lemma A.1 that the removal of the $[Y_i]$-cycle can be done through the companion states $(Y_i^{q_i} 0^{r_i} 1)$ and $(Y_i^{q_i} 0^{r_i+1})$, where q_i and r_i are defined in (6).

The situation can be simplified for i such that 2^i divides n.

A $[Y_i]$-cycle is a PCR_n-cycle iff 2^i is a divisor of n. Hence, if $l_i = 1$ and 2^i divides n, the removal of the $[Y_i]$-cycle can be saved if it is not joined to the MC to begin with. That is, in the process of forming the final MC, we can skip those $[Y_i]$-cycles for which 2^i divides n and $l_i = 1$. There is a virtual inconsistency here, as the l_i's are determined from the difference $L' - L$, where L' is computed with the understanding that no cycles are skipped in the construction process. However, since skipping such a cycle has the same effect as first putting it in and then removing it, we may as well skip it. The same is true for the cycle [0] if $l_0 = 1$. In this case, we take $[0^{n-1}1]$ as the initial MC.

2.4. *Alternate selections of bridging states*

The construction described above generates one L-cycle. Our purpose, however, is to suggest an algorithm for the production of a large class of L-cycles. This can be achieved by providing a multiple choice of possible bridging states for the various PCR_n-cycles. Given a PCR_n-cycle C and a state X of C, X can serve as an alternate bridging state for C if all of the following five properties hold (Lemma A.10):

(P1) The last bit of X is one.

(P2) X' is not on a $[Y_i]$-cycle, $1 \leq i < \log n$.

(P3) C is not a member of the exception set.

(P4) C is not of the form $[0^{r_i+2^i} 1 Y_i^{q_i-1}]$, where i, r_i, and q_i satisfy (6).

(P5) C is not of the form $[10^k 10^{n-k-3}1]$, $n - 2^{\lfloor \log n \rfloor} \leq k \leq \frac{1}{2}(n-2)$, if n is not a power of 2.

We claim that each state X of the set V defined below (which is similar to the sets in [1, 2]) and the cycle C to which X belongs satisfy (P1) through (P5). Let $n \geq 13$ and let $V = \{V(i)\}_{i=0}^{u-1}$ be the u states, $4 \leq u \leq 2^{(n-\lfloor n/2 \rfloor - 3)/2}$, constructed as follows:

(V1) The first $\lceil \log u \rceil + 1$ bits of $V(i)$ form the base-2 representation of i (note, therefore, that the first bit is always zero).

(V2) The last $\lfloor \log u \rfloor + 1$ of each $V(i)$ are ones.

(V3) The $\lfloor \frac{1}{2}n \rfloor$ bits preceding the last $\lfloor \log u \rfloor + 1$ bits are zeros.

(V4) For each $V(i) = (V_1(i), V_2(i), \ldots, V_n(i))$, the bits in positions $\lceil \log u \rceil + 2 + (\lfloor \log u \rfloor + 1)j$, where

$$0 \leq j < \left\lceil \frac{n - 2 - \lfloor \frac{1}{2}n \rfloor - \lceil \log u \rceil - \lfloor \log u \rfloor}{\lfloor \log u \rfloor + 1} \right\rceil,$$

are zeros.

(V5) The remaining bits of each $V(i)$ are free parameters that can be fixed arbitrarily. Different assignments for these bits define different sets V.

Our claim regarding the set V is based on the arguments given after the following example.

Example 2.4. $n = 21$, $u = 4$. The parametric form of the set V for these values of n and u is:

$$0000\,V_5(0)\,V_6(0)0\,V_8(0)0000000000111,$$

$$0010\,V_5(1)\,V_6(1)0\,V_8(1)0000000000111,$$

$$0100\,V_5(2)\,V_6(2)0\,V_8(2)0000000000111,$$

$$0110\,V_5(3)\,V_6(3)0\,V_8(3)0000000000111.$$

The total number of distinct sets V for this example is $(2^3)^4 = 4096$. Each of these sets gives rise to a different L-cycle and, thus, 4096 is the number of distinct L-cycles that can be generated by the proposed algorithm when $n = 21$ and $u = 4$.

First, by (V1) and (V3), it is clear that each state of any specific set V will be on every constructed L-cycle, since the weight of each state in V is less than $\frac{1}{2}n \leq m$, the final weight.

Second, it has to be shown that each state of a specific set V belongs to a different PCR_n-cycle. This follows from the observation that the run of the rightmost $\lfloor \log u \rfloor + 1$ ones forms the unique longest run of ones in each $V(i)$, and that every pair of states differ in their first $\lceil \log u \rceil + 1$ bits.

Finally, it is easy to verify that properties (P1) through (P5) follow from (V2) which implies that each member of V has a run of at least three ones. More specifically, (P1) and (P2) directly follow; (P3), (P4), and (P5) hold for any cycle containing a state with a run of at least three ones.

We summarize this in the form of Theorem 2.5 given below, which directly follows from the lemmas and the discussion in Subsections 2.1 through 2.4.

Theorem 2.5. *Given a specific set V, the construction described in Subsections* 2.1–2.4 *produces an L-cycle.*

3. A bit-by-bit algorithm

The construction of L-cycles proposed in Section 2 was described in terms of smaller constituent cycles—PCR_n-cycles and $[Y_i]$-cycles—and the techniques of joining or removing such cycles to obtain the desired output.

In practice, it is preferable to have an algorithm that produces the bits of an L-cycle one-by-one, in their natural order in the desired L-cycle, where the production of the next bit may depend on the last few bits already produced and on some sorted data.

Algorithm A given below does just that by translating the construction described in the previous section into a bit-by-bit algorithm. The value of the next bit on the cycle produced by this algorithm is determined by the following inputs.

Inputs for Algorithm A

(I1) The current state $\beta_i = (b_i, b_{i+1}, \ldots, b_{i+n-1})$ and, specifically, the value of b_i and the weight w_i of β_i. (The initial state is $(0^{n-1}1)$.)

(I2) The number f_i of PCR_n-cycles of final weight m and final length h for which the current output contains at least one of their states.

(I3) The parameter g_i which equals 1, unless $m = \frac{1}{2}(n+1)$, $h = n$, and no state of the cycle $[(10)^{(n-1)/2}1]$ is in the current output in which case $g_i = 0$.

(I4) Stored data consisting of: the exception set of PCR_n-cycles along with selected states from these cycles, a fully specified set V of bridging states, and the precomputed values of the parameters m, h, t, and l_i, $0 \leq i < \log n$.

More specifically, the stored inputs to the algorithm are determined as follows.

Stored data for Algorithm A

(D1) Let 2^{i_0} be the highest power of 2 that divides n. For each i such that $i_0 < i < \log n$, store:

$$U(i,1) = (0^{2^i-1}10^{r_i}1\,Y_i^{q_i-1}), \qquad U(i,2) = (0^{r_i}1\,Y_i^{q_i}), \qquad U(i,3) = (Y_i^{q_i}0^{r_i}1),$$

where q_i and r_i are defined in (6).

(D2) If n is not a power of 2, then, for each k such that $n - 2^{\lfloor \log n \rfloor} \leq k \leq \frac{1}{2}(n-2)$ store: $R(k) = (10^k10^{n-k-3}1)$.

(D3) Store a fully specified set $V = \{V(i)\}_{i=0}^{u-1}$ satisfying (V1) through (V5).

(D4) Store the precomputed values of the final weight m, the final length h and the number t (see inequalities (1), (4) and (5)).

(D5) Store the precomputed values of l_i, $0 \leq i < \log n$ (see (3)).

Given the current state $\beta_i = (b_i, b_{i+1}, \ldots, b_{i+n-1})$, the formal steps by which the next bit b_{i+n} of the L-cycle is determined are presented in Algorithm A below.

Algorithm A. Initially, set $\beta_0 = (0^{n-1}1)$, $w_0 = 1$, $f_0 = 0$, and

$$g_0 = \begin{cases} 0 & \text{if } m = \frac{1}{2}(n+1) \text{ and } h = n, \\ 1 & \text{otherwise.} \end{cases}$$

Given $\beta_i = (b_i, b_{i+1}, \ldots, b_{i+n-1})$, w_i, f_i, and g_i, let $\beta_i^* = (b_{i+1}, \ldots, b_{i+n-1}, 1)$ and let C_i^* be the PCR_n-cycle containing the state β_i^*. Proceed to produce $\beta_{i+1} = (b_{i+1}, \ldots, b_{i+n-1}, b_{i+n})$, w_{i+1}, f_{i+1}, and g_{i+1} as follows:

(A1) If $\beta_i = (10^{n-1})$, set $b_{i+n} = w_{i+1} = l_0$, $f_{i+1} = 0$,

$$g_{i+1} = \begin{cases} 0 & \text{if } m = \frac{1}{2}(n+1) \text{ and } h = n, \\ 1 & \text{otherwise.} \end{cases}$$

and stop;
if $\beta_i \neq (10^{n-1})$ and n is not a power of 2 go to (A5).

(A2) If $m = \frac{1}{2}(n+1)$ and $h = n$, then:
- if $l_1 = 0$ and $\beta_i = (0(10)^{(n-1)/2})$, go to (A10);
- if $l_1 = 1$ and $\beta_i = (0(01)^{(n-1)/2})$, set $b_{i+n} = g_{i+1} = 1$, $w_{i+1} = \frac{1}{2}(n+1)$, $f_{i+1} = f_{i+1}$, and stop.

(A3) If $S^*(C_i^*) \neq U(j, 1)$ for all j, go to (A4);
if $S^*(C_i^*) = U(j, 1)$ for some j and either $\beta_i^* = U(j, 2)$, or $\beta_i^* = U(j, 3)$ and $l_j = 1$, go to (A9);
otherwise go to (A8).

(A4) If $\beta_i^* = R(k)$, for some k, go to (A9);
if $\beta_i^* = S^*(C_i^*)$ and the companion of β_i^* belongs to the $[Y_{\lfloor \log n \rfloor}]$-cycle, go to (A8).

(A5) If C_i^* is a $[Y_j]$-cycle, then:
- if $l_j = 1$, go to (A8), and
- if $l_j = 0$, go to (A9).

(A6) Look for a state α on C_i^* that begins with a zero and ends with $1 + \lfloor \log u \rfloor$ ones. If no such α exists go to (A7). Let α^* be the first $1 + \lfloor \log u \rfloor$ bits of α and let $k = |\alpha^*|$, the base-2 value of α^*. If $k > u - 1$, go to (A7); otherwise, if $\alpha = V(k) = \beta_i^*$, go to (A9) and if $\alpha = V(k) \neq \beta_i^*$, go to (A8).

(A7) Let $\lambda(C_i^*)$ denote the length of C_i^*. If $\beta_i^* = S^*(C_i^*)$ and
- if $w_i < m - 1$ or $b_i = 1$, go to (A9);
- if $w_i = m - 1$ and $b_i = 0$, then:
  - (a) if $\lambda(C_i^* > h$, go to (A9);
  - (b) if $\lambda(C_i^*) < h$, go to (A8);
  - (c) if $\lambda(C_i^*) = h$ and $f_i = t$, go to (A8);
  - (d) if $\lambda(C_i^*) = h$ and $f_i < t - 1$, go to (A10);
  - (e) if $\lambda(C_i^*) = h$ and $f_i = t - 1$, then:
    if $g_i = 1$, go to (A10), and
    if $g_i = 0$, go to (A8).

(A8) Set $b_{i+n} = b_i$, $w_{i+1} = w_i$, $f_{i+1} = f_i$, $g_{i+1} = g_i$ and stop.

(A9) Set $b_{i+n} = \bar{b}_i$, $w_{i+1} = w_i + b_{i+n} - b_i$, $f_{i+1} = f_i$, $g_{i+1} = g_i$, and stop.

(A10) Set $b_{i+n} = \bar{b}_i 1$, $w_{i+1} = w_i + b_{i+n} - b_i$, $f_{i+1} = f_i + 1$, $g_{i+1} = g_i$, and stop.

The following theorems are proved in Appendix A.

Theorem 3.1. *Given the current state, Algorithm* A *produces the next bit of an L-cycle.*

Theorem 3.2. (a) *Given* u, $4 \leq u \leq 2^{(n-\lfloor n/2 \rfloor - 3)/2}$, *there are* $2^{ud(n,u)}$ *distinct choices for the set* V, *where*

$$d(n,u) = n - 2 - \lfloor \tfrac{1}{2}n \rfloor - \lceil \log u \rceil - \lfloor \log u \rfloor - \left\lceil \frac{n - 2 - \lfloor \frac{1}{2}n \rfloor - \lceil \log u \rceil - \lfloor \log u \rfloor}{\lfloor \log u \rfloor + 1} \right\rceil.$$

Thus, Algorithm A *can be used to produce* $2^{ud(n,u)}$ *distinct L-cycles.*

(b) *The* $2^{ud(n,u)}$ *L-cycles that Algorithm* A *produces can be generated in lexicographical order*

(c) *The number of* (n-*bit*) *operations that Algorithm* A *requires to produce an output bit is* $o(n)$.

(d) *The working space that Algorithm* A *requires is* $o(\max\{n^2, ud(n,u)\})$.

The present author [1] gave an algorithm to construct m-ary De Bruijn sequences. A combination of this algorithm and Algorithm A can produce an algorithm to generate cycles of length L, $1 \leq L \leq m^n$, such that no m-ary n-tuple appears twice in a cycle, for any L, m, and n.

Appendix A

Proof of Theorem 3.1. The validity of Algorithm A follows from Theorem 2.5. To see this, it suffices to show that, given the current state $\beta_i = (b_i, b_{i+1}, \ldots, b_{i+n-1})$, the next bit b_{i+n} is set equal to b_i or $\bar{b}_i$ depending on whether β_i and β_{i+1} are on the same PCR_n-cycle or two different PCR_n-cycles, respectively. In the construction of Section 2 there are only two situations in which β_{i+1} is not on the same PRC_n-cycle as β_i. These situations arise when:

(1) in the process of forming the final MC, either β_{i+1} or its companion β'_{i+1} serves as a bridging state;

(2) β_{i+1} and β'_{i+1} are a pair of companion states which serve to remove a $[Y_j]$-cycle from the final MC. That is, β_{i+1} is of the form $(Y_j^{q_j} 0^{r_j} 1)$ for some j with $l_j = 1$. In the sequel, such a state β_{i+1} will be referred to as a *severing* state; bridging or severing states will be referred to as *critical* states.

To determine whether the current state of the algorithm corresponds to one of the above situations, we check whether $\beta_i^* = (b_{i+1}, b_{i+2}, \ldots, b_{i+n-1}, 1)$ or its companion serves as a critical state. This is exactly what is done in most of the steps of Algorithm A.

In addition, we have to show that the updating of the parameters w_i, f_i, and g_i is consistent with their definitions (see (I1), (I2), and (I3)). The updating of w_i is simple and its verification is left to the reader. For the proper updating of f_i and g_i we need to detect the first occurrence in the output of a state which belongs to a PCR_n-cycle of final weight m and final length h. To this end we apply Lemma A.12 as needed.

The first two steps, (A1) and (A2), deal with the specific questions regarding the PCR_n-cycles [0] and [01] that can directly be answered without reference to β_i^*.

The validity of step (A1) follows from the observation that the state (10^{n-1}) is followed on the L-cycle either by (0^n) or by $(0^{n-1}1)$, depending on whether $l_0=0$ or $l_0=1$, respectively (see Subsection 2.3). The set of values for f_{i+1} and g_{i+1} are set equal to the initial ones, since (10^{n-1}) and (0^n) are the possible predecessors of the initial state $(0^{n-1}1)$.

To validate step (A2), we recall that if $m=\frac{1}{2}(n+1)$ and $h=n$, then the PCR_n-cycle $C_1^2=[(10)^{(n-1)/2}1]$ (see Subsection 2.2) is the first cycle of the cited weight and length to join the MC (see Subsection 2.1). By Lemma A.12, the first state in the output that belongs to C_1^2 is either the bridging state $((10)^{(n-1)/2}1)$ of C_1^2 when $l_1=0$, or the severing state $((01)^{(n-1)/2}1)$ of $[Y_1]=[01]$ when $l_1=1$. The jump to (A10) here, as well as jumps to (A8), (A9), and (A10) from the succeeding steps should be obvious from the conditions under which such jumps are made.

To validate step (A3), assume $S^*(C_i^*)=U(j,1)$ for some j. Then either β_{i+1} or β'_{i+1} is a state that belongs to the cycle C_j^2. If $\beta_i^*=U(j,2)$, then either β_{i+1} or β'_{i+1} is the bridging state $U(j,2)$; if $\beta_i^*=U(j,3)$ and $l_j=1$, then β_{i+1} serves as the severing state $U(j,3)$ of $[Y_j]$. If neither $\beta_i^*=U(j,2)$ nor $\beta_i^*=U(j,3)$ and $l_j=1$, then β_i^* is not a critical state.

In step (A4) we consider the PCR_n-cycles of the form: $Z_k=[0^{n-k-2}10^k1]$, $n-2^{\lfloor \log n \rfloor} \leqslant k < \frac{1}{2}(n-2)$, and $Z_{(n-2)/2}=[0^{(n-2)/2}1]$ when n is even. If $\beta_i^*=R(k)$, then it is the bridging state of Z_k (see Subsection 2.2). Assume now that $\beta_i^*=S^*(C_i^*)$ and that the companion of β_i^* belongs to the $[Y_{\lfloor \log n \rfloor}]$-cycle. Since $C_i^* \neq C_{\lfloor \log n \rfloor}^2$ (cycles of the form C_j^2 are treated in (A3)), it follows from Lemma A.4 that β_i^* is on a Z_k-cycle. Since the Z_k-cycles belong to the exception set, their bridging states are not of the form $S^*(Z_k)$ and, hence, β_i^* is not a critical state.

Step (A5) treats $[Y_j]$-cycles such that 2^j divides n, i.e., $[Y_j]$-cycles that are PCR_n-cycles. It is clear that, for such j, $S^*([Y_j])$ is the state of $[Y_j]$ that ends in a one. Therefore, assuming $C_i^*=[Y_j]$, we have $\beta_i^*=S^*([Y_j])$. If $l_j=0$, β_i^* is a bridging state; if $l_j=1$, $[Y_j]$ is not joined to the MC to begin with (see Subsection 2.3) and, hence, β_i^* is not a crical state.

The validity of (A6) follows from the fact that each $V(k)$ is a bridging state on some PCR_n-cycle C while all the other states of C are not critical ones (see Subsection 2.4).

The validity of (A7) follows from the following observations:

(1) For each PCR_n-cycle C which is not treated in Steps (A1) through (A6), the bridging state is taken as $S^*(C)$ (see Subsection 2.2).

(2) The cycles of final weight m are taken in order of decreasing length with h being the final length (see Subsection 2.1).

(3) The cycle $[(10)^{(n-1)/2}1]$ must be joined to the MC.

(4) The first state in the output of Algorithm A that belongs to a PCR_n-cycle C, $C \neq [(10)^{(n-1)/2}1]$, of weight m and length h is $S^*(C)$, the bridging state of C (see Lemma A.12).

All the cases in which either β_i^* or its companion is a critical state are treated in steps (A1) through (A7) and steps (A9) and (A10). Every case where either β_i^* nor its companion is a critical state is treated in (A8). □

Proof of Theorem 3.2. Part (a) is valid because each $V(i)$ is specified up to exactly $d(n, u)$ free parameters and because distinct sets V of alternate bridging states yield distinct L-cycles. The latter is due to the fact that except for $(0^{n-a}1^a)$, $a = 1 + \lfloor \log u \rfloor$, no state of the form $S^*(C)$ ever serves as a member of the set V.

Part (b) follows from the observation that, by (V5) of Subsection 2.4, the $ud(n, u)$ free parameters of the set V can get their assignment in lexicographical order.

Part (c) follows from an inspection of the steps of Algorithm A. Steps (A1), (A2), (A8), (A9), and (A10) need only a constant number of n-bit operations. In some of the other steps we have to find $S^*(C_i^*)$ which requires no more than n cyclic shifts and the same number of n-bit comparisons.

Step (A3) requires no more than $\log n$ comparisons to check whether $S^*(C_i^*) = U(j, 1)$ for some j.

In step (A4) we need no more than n comparisons to check whether $\beta_i^* = R(k)$ for some k and less than n cyclic shifts and n comparisons to check whether the companion of β_i^* belongs to $[Y_{\lfloor \log n \rfloor}]$.

Step (A5) requires no more than n comparisons if the $[Y_j]$ are kept in storage or n cyclic shifts and n comparisons if the $[Y_j]$ are not stored. In order to find a state beginning with a zero and ending with $1 + \lfloor \log u \rfloor$ ones, if one exists, (A6) requires at most n cyclic shifts and at most n comparisons.

In step (A7) we have to find the length of C_i^*, which is done by finding the period of β_i^*, i.e., the smallest number of cyclic shifts under which β_i^* is left unchanged. For this we need at most n cyclic shifts and the same number of comparisons. All other operations in (A7) require a constant time of operations.

Summing up, it follows that, Algorithm A requires $o(n)$ operations to produce an output bit.

To verify (d), we check the storage needed for the inputs (I1) through (I4). As far as the set V is concerned, on the $d(n, u)$ free parameters of each $V(i)$ require storage. Hence, storing the set V requires $ud(n, u)$ bits. It is also clear that the other inputs require no more than $o(n^2)$ bits. □

Lemma A.1. *Let n, i, q_i, and r_i be integers such that $n - 1 = 2^i q_i + r_i$, where i, $q_i > 0$ and $0 \leq r_i \leq 2^i - 2$. Then the PCR_n-cycles*

$$C_i^1 = [0^{r_i+2^i} 1 Y_i^{q_i-1}], \qquad C_i^2 = [0^{r_i} 1 Y_i^{q_i}],$$

can be joined into one cycle C, which, in turn, can be split into the pair of cycles $[Y_i]$ *and* C_i^3 *with* $S^*(C_i^1)$ *on* C_i^3.

Proof. C_i^1 and C_i^2 can be joined by applying Theorem 2.1 to the state $X=(0^{r_i}1Y_i^{q_i})$ on C_i^2 and its companion $X'=(0^{r_i}y_i^{q_i-1}0^{2^i})$ on C_i^1, to obtain the cycle

$$C=[0^{r_i}1Y_i^{q_i-1}0^{2^i}0^{r_i}1Y_i^{q_i}]=[Y_i^{q_i-1}0^{r_i+1}Y_iY_i^{q_i-1}0^{r_i+1}0^{2^i-r_i-2}10^{r_i}1].$$

Applying Theorem 2.1 to the state $(Y_i^{q_i}0^{r_i}1)$ and its companion on C we obtain the cycles $[Y_i]$ and $C_i^3=[Y_i^{q_i-1}0^{r_i+2^i}1Y_i^{q_i-1}0^{r_i}1]$. One can easily verify that $S^*(C_i^1)=(0^{r_i+2^i}1Y_i^{q_i-1})$ is on C_i^3. □

Corollary A.2. *Assume that* C_i^1 *is joined to the MC, with* $S^*(C_i^1)$ *as the bridging state. Assume further that the contiguity of* $[Y_i]$*-states is preserved on the MC until* C_i^2 *is to be joined to it. Then, after joining* C_i^2 *with* $(0^{r_i}1Y_i^{q_i})$ *as the bridging state, all the states of* $[Y_i]$ *are on the MC and are contiguous on it.*

Corollary A.3. *If all the states of* $[Y_i]$ *are contiguous on the MC, then* $[Y_i]$ *can be removed by applying Theorem* 2.1 *to the state* $(Y_i^{q_i}0^{r_i+1})$ *and its companion.*

Lemma A.4. *Let i and n be such that* $0<i<\log n$ *and* 2^i *does not divide n. Let C be a* PCR_n*-cycle,* $C\neq C^2_{\lfloor\log n\rfloor}$. *If the companion of* $S^*(C)$ *is a state of* $[Y_i]$, *then* $i=\lfloor\log n\rfloor$ *and C is either of the form* $[0^{n-k-2}10^k1]$, $n-2^{\lfloor\log n\rfloor}\leq k<\frac{1}{2}(n-2)$, *or* $C=[0^{(n-2)/2}1]$.

Proof. Let j, q_{ij}, and r_{ij} be integers such that $0<j\leq 2^i-1$ and $n-j-1=2^iq_{ij}+r_{ij}$, where $q_{ij}\geq 0$ and $0\leq r_{ij}\leq 2^i-1$. The 2^i states of $[Y_i]$ are $(0^{r_{ij}}1Y_i^{q_{ij}}0^j)$, $0\leq j\leq 2^i-1$. Hence, if the companion $S^*(C)$ is a state on $[Y_i]$, then $S^*(C)$ is of the form $(0^{r_{ik}}1Y_i^{q_{ik}}0^{k-1}1)$ for some k, $1\leq k\leq 2^i-1$, and $r_{ik}>0$ (recall that, by definition, $S^*(C)$ ends with a one, and if $C\neq[0]$, $S^*(C)$ begins with a zero). We claim that $q_{ik}=0$. Assuming $q_{ik}>0$, we can write $S^*(C)=(0^{r_{ik}}1Y_i^{q_{ik}-1}0^{2^i-1}10^{k-1}1)$, or $S(C)=(1Y_i^{q_{ik}-1}0^{2^i-1}10^{k-1}10^{r_{ik}})$. From the definition of $S(C)$, it follows that $2^i-1\leq k-1$ which contradicts the defined range of k. Hence, $q_{ik}=0$ and $S^*(C)=(0^{r_{ik}}10^{k-1}1)$. Since $2^i-1>r_{ik}$ and $2^i-2\geq k-1$, it follows that $n<2^{i+1}$. From the assumed range of i, it follows that $n>2^i$ which, combined with $n<2^{i+1}$, implies $i=\lfloor\log n\rfloor<\log n$.

Now, since $S^*(C)$ is of length n, we can write $S^*(C)=(0^{n-k-2}10^k1)$ for some k, $0\leq k\leq 2^i-2$. It remains to be shown that, actually, k is restricted to the range stated in Lemma A.4, or that $C=[0^{(n-2)/2}1]$. The stated upper bound on k follows from the inequality $k\leq n-k-2$ as required by the definition of $S^*(C)$ (the case $k=n-k-2$ yields $C=[0^{(n-2)/2}1]$). On the other hand, being a companion of a state on $[Y_i]$, the length of every run of zeros in $S^*(C)$ cannot exceed 2^i-1. Therefore, $n-k-2\leq 2^i-1=2^{\lfloor\log n\rfloor}-1$. From the definition of C_i^2 in Lemma A.1, we have

$$C^2_{\lfloor\log n\rfloor}=[0^{r_{\lfloor\log n\rfloor}}1Y_{\lfloor\log n\rfloor}]=[Y_{\lfloor\log n\rfloor}0^{r_{\lfloor\log n\rfloor}}1]=[0^{2^{\lfloor\log n\rfloor}-1}10^{n-2^{\lfloor\log n\rfloor}-1}1].$$

Since we assumed $C\neq C^2_{\lfloor\log n\rfloor}$, it follows that $n-k-2\neq 2^{\lfloor\log n\rfloor}-1$ and, thus, $n-k-2<2^{\lfloor\log n\rfloor}-1$, or $k\geq n-2^{\lfloor\log n\rfloor}$ as claimed. □

One can easily verify the following lemma.

Lemma A.5. *Let $X_0, X_1, \ldots, X_k$ be $k+1$ contiguous states on the MC. Let C be a PCR_n-cycle, which is joined to the MC by applying Theorem 2.1 to state Y of C, and state Y' of the MC. Then the states $X_0, X_1, \ldots, X_k$ remain contiguous on the MC iff $Y' \notin \{X_1, \ldots, X_k\}$.*

Corollary A.6. *Let n and i be integers such that $0 < i < \log n$ and 2^i does not divide n. Let C be a PCR_n-cycle. Assume that either $i < \lfloor \log n \rfloor$ or $C \neq [0^{(n-2)/2}1]$ and C is not of the form $[0^{n-k-2}10^k1]$, $n - 2^{\lfloor \log n \rfloor} \leq k < \frac{1}{2}(n-2)$. Assume further that the $[Y_i]$-states belonging to the current MC are contiguous on it. If C is joined to the current MC through the bridging state $S^*(C)$ or through $(0^{r_j}1Y_j^{q_j})$ if $C = C_j^2$, then the $[Y_i]$-states remain contiguous on the main MC.*

Proof. From Lemmas A.4 and A.5 it follows that if $C \neq C^2_{\lfloor \log n \rfloor}$ and if $S^*(C)$ is taken as the bridging state of C, then only the contiguity of $[Y_{\lfloor \log n \rfloor}]$ might be destroyed. Furthermore, this happens only if C is of the form $[0^{n-k-2}10^k1]$, $n - 2^{\lfloor \log n \rfloor} \leq k < \frac{1}{2}(n-2)$ or $C = [0^{(n-2)/2}1]$. By Corollary A.2, the contiguity is preserved when $(0^{r_j}1Y_j^{q_j})$ is taken as the bridging state of C_j^2. □

Lemma A.7. *Let n and i be integers such that $i < \log n$ and 2^i is a divisor of n. If C is a PCR_n-cycle, then the companion of $S^*(C)$ is not a state of the cycle $[Y_i]$.*

Proof. *Let* $q = (n/2^i) - 1$. The 2^i states of $[Y_i]$ are $(0^j1Y_i^q0^{2^i-j-1})$, $0 \leq j \leq 2^i - 1$. Hence, the companion of $S^*(C)$ is of the form $(0^k1Y_i^q0^{2^i-k-2}1)$ for some k, $0 \leq k \leq 2^i - 2$. By an argument similar to that in the proof of Lemma A.4, this implies that $q = 0$ and $i = \lfloor \log n \rfloor$. Since 2^i divides n, it follows that $i = \log n$, which contradicts the assumption $i < \log n$. Hence, the claim in the lemma is obtained. □

Corollary A.8. *Let n and i be integers such that $i < \log n$ and 2^i is a divisor of n. Assume that the $[Y_i]$-states that belong to the current MC are contiguous. If a PCR_n-cycle C is joined to the current MC by applying Theorem 2.1 to the states $S^*(C)$ and its companion, then the $[Y_i]$-states remain contiguous on the new MC.*

Lemma A.9. *Consider the PCR_n-cycles of the form: $Z_k = [0^{n-k-2}10^k1]$, $n - 2^{\lfloor \log n \rfloor} \leq k < \frac{1}{2}(n-2)$, and $Z_{(n-1)/2} = [0^{(n-2)/2}1]$ when n is even. If n is not a power of 2, then the MC that consists of all the PCR_n-cycles of weight less than four can be formed by joining these cycles via the following bridging states:*

(1) (10^k10^{n-k-2}) *for each Z_k-cycle,*

(2) $(0^{n-2^{\lfloor \log n \rfloor}-1}10^{2^{\lfloor \log n \rfloor}-1}1)$ *for the cycle $C^2_{\lfloor \log n \rfloor} = [0^{n-2^{\lfloor \log n \rfloor}-1}10^{2^{\lfloor \log n \rfloor}-1}1]$,*

(3) $S^*(C)$ *for the remaining PCR_n-cycles.*

As a result, in the final MC of length L', all the states of $[Y_{\lfloor \log n \rfloor}]$ are contiguous.

Proof. Initially, the MC consists of the PCR_n-cycle [0]. Next, $C=[0^{n-1}1]$ is joined via $S^*(C)$. Then the cycle $C^2_{\lfloor \log n \rfloor}=[0^{n-2^{\lfloor \log n \rfloor}-1}10^{2^{\lfloor \log n \rfloor}-1}1]$ can be joined to the MC through the bridging state $(0^{r_{\lfloor \log n \rfloor}}1\,Y^{q_{\lfloor \log n \rfloor}}_{\lfloor \log n \rfloor})=(0^{n-2^{\lfloor \log n \rfloor}-1}10^{2^{\lfloor \log n \rfloor}-1}1)$. Now, the other PCR_n-cycles C of weight two, except for the Z_k-cycles, are joined to the MC through $S^*(C)$. Since $n-2^{\lfloor \log n \rfloor}\leq k\leq \frac{1}{2}(n-2)$, $n>4$, and n is not a power of 2, we have $k>0$ and $n-k-3>0$. Hence, the PCR_n-cycle $[110^{n-2}]$ is not a Z_k-cycle and therefore, its states are already on the current MC. Now, the cycles C of the form $[10^k10^{n-k-3}1]$, for each k in the given range, are joined to the MC via $S^*(C)=(0^{n-k-3}110^k1)$ (the companion of $S^*(C)$ is a state on the cycle $[110^{n-2}]$). Next, the Z_k-cycles are joined to the MC via the states $(0^{n-k-2}10^k1)$ as the corresponding bridging states. The remaining PCR_n-cycle of weight 3 can be joined to the MC through $S^*(C)$.

As a result of the fact that the companion of each bridging state of a Z_k-cycle is a state on the cycle $[110^{n-2}]$, it follows by Lemmas A.1, A.5 and Corollary A.6, that the states belonging to the cyclee $[Y_{\lfloor \log n \rfloor}]$ are contiguous. □

Lemma A.10. *Let C be a PCR_n-cycle and let X be a state on C. X can serve as an alternate bridging state of C if properties* (P1) *through* (P5) *hold.*

Proof. When it is the turn of C to join the MC, X can serve as a bridging state for the join if X' is then on the MC and the join preserves the contiguity of $[Y_i]$-states on the MC.

The presence of X' on the MC is guaranteed by (P1) since the PCR_n-cycles are joined to the MC in order of increasing weight. The only exception to this rule are the Z_k-cycles (see Lemma A.9) which are anyway excluded by (P3). Preservation of contiguity is guaranteed by (P2) through (P5). □

Lemma A.11. *If an MC is produced by a sequential bit-by-bit algorithm and the initial state is taken as $(0^{n-1}1)$, then the first state of a PCR_n-cycle C that appears in the output is the bridging state of C.*

Proof. The lemma is proved by induction on the number of PCR_n-cycles that have been joined to the MC.

Basis step: If the MC consists of only two PCR_n-cycles, [0] and $[0^{n-1}]$, the lemma is obvious.

Induction step: Assume the claim holds for an MC consisting of k PCR_n-cycles. Let M_2 be an MC which consists of $k+1$ PCR_n-cycles. M_2 is produced by joining some PCR_n-cycle $C=[X_1, X_2, \ldots, X_l]$ to M_1 which is an MC consisting of k PCR_n-cycles. By the induction hypothesis, the claim is true for M_1. Let X_i be the bridging state of C. X'_i is on M_1 and let Y be the predecessor of X'_i on M_1. Applying Theorem 2.1 to the states X_i and X'_i will produce M_2 with the states Y, X_i, $X_{i+1}, \ldots, X_l$, $X_1, \ldots, X_{i-1}$, X'_i contiguous and in this order on M_2. The relative order among the states of M_1 remains unchanged on M_2. Hence, the first state of

C in the output is its bridging state X_i and the first state of any other PCR_n-cycle contained in M_2 is the same as in M_1. □

Note that Lemma A.11 is valid for each MC, including the final one. The following lemma applies to the case where a bit-by-bit algorithm is used all the way to produce the desired L-cycle (which is obtained from the final MC after the removal of certain $[Y_i]$-cycles).

Lemma A.12. *Assume that the L-cycle produced in Section 2 is obtained by a sequential bit-by-bit algorithm and that the initial state is taken as* $(0^{n-1}1)$. *Then the first state* S *of a* PCR_n*-cycle* C *in the output of the algorithm is the bridging state of* C *unless* C *contains a state of the form* $(Y_i^{q_i}0^{r_i}1)$ *for some* i *and* $l_i = 1$. *In this case* $S = (Y_i^{q_i}0^{r_i}1)$.

Proof. If no $[Y_i]$-cycle is removed, all $l_i = 0$ and the claim follows from Lemma A.11. Assume that $i_j = 1$ for some j and that $[Y_j] = [X_1, X_2, \ldots, X_{2^j}]$. Consequently, $[Y_j]$ is removed from the final MC. When we recall that the states of $[Y_j]$ are contiguous on the final MC and that it can be removed by a single application of Theorem 2.1 (see Lemma A.1), the cycle $[Y_j]$ and its neighbouring states on the final MC must take the form $Y, X_1, X_2, \ldots, X_k, X_{k+1}, \ldots, X_{2^j}, X_1'$, where $X_1 = (Y_j^{q_j}0^{r_j+1})$, X_1 through X_k are states on C_j^1 and X_{k+1} through X_{2^j} are states on C_j^2. Furthermore, $X_{k+1} = (0^{r_j}1\,Y_j^{q_j})$ is the bridging state of C_j^2 and none of the X_i's is the bridging state of C_j^1. Hence, when $[Y_j]$ is removed via X_1 and X_1', the latter becomes the first state of C_j^2 in the output and for each other PCR_n-cycle C the first state in the output remains unchanged. □

Acknowledgment

The author wishes to thank Professor Abraham Lempel for many helpful discussions and presentation of the paper.

References

[1] T. Etzion, An algorithm for constructing m-ary De Bruijn sequences, *J. of Algorithms*, to appear.
[2] T. Etzion and A. Lempel, Algorithms for the generation of full-length shift-register sequences, *IEEE Trans. Inform. Theory* **IT-30** (1984) 480–484.
[3] S. Even, *Algorithmic Combinatorics* (Macmillan, New York, 1973).
[4] S.W. Golomb, *Shift Register Sequences* (Aegean Park Press, Laguna Hills, 1982).
[5] H.M. Fredriksen, A class of nonlinear De Bruijn cycles, *J. Combin. Theory Ser. A* **12** (1975) 192–199.
[6] F. Hemmati and D.J. Costello, An algebraic construction for q-ary shift register sequences, *IEEE Trans. Comput.* **G-27** (1978) 1192–1195.
[7] A. Lempel, m-Ary closed sequences, *J. Combin. Theory Ser. A* **10** (1971) 253–258.