De Bruijn Sequences from Symmetric Shift Registers

Ming Li, Mingxing Wang and Dongdai Lin

State Key Laboratory of Information Security,
Institute of Information Engineering,
Chinese Academy of Sciences, Beijing 100093, China
E-mail: {liming,wangmingxing,ddlin}@iie.ac.cn

November 6, 2015

Abstract

We consider the symmetric Feedback Shift Registers (FSRs), especially a special class of symmetric FSRs (we call them scattered symmetric FSRs), and construct a large class of De Bruijn sequences from them. It is shown that, at least $O(2^{\frac{n-6}{2}\log n})$ De Bruijn sequences of order n can be constructed from just one n-stage scattered symmetric FSR. To generate the next bit in the De Bruijn sequence from the current state, it requires no more than $2n$ comparisons and $n+1$ FSR shifts. By further analyse the cycle structure of the scattered symmetric FSRs, other methods for constructing De Bruijn sequences are suggested.

Keywords: symmetric boolean function, feedback shift register, De Bruijn sequence, cycle joining method.

1 Introduction

De Bruijn sequences, i.e., periodic sequences in which each n-tuple appears exactly once in one period, have been studied for many years, see, for example, [1, 4]. These sequences have many applications in cryptography and modern communication systems. Numerous algorithms for generating these sequences are known, and a useful survey has been given by Fredricksen [4].

A classical method to construct De Bruijn sequences is to consider a feedback shift register (FSR) producing several short cycles which are then joined together to form a full cycle. Linear feedback shift registers (LFSRs) with simple cycle structures are often used for this purpose, for example, the maximum length LFSRs, pure circulating registers and pure summing registers [2–4]. Recently, the LFSRs with characteristic polynomials $(1+x)^m p(x)$ and $(1+x^m)p(x)$ were also used, where $p(x)$ is a primitive polynomial and m is a positive integer [7, 10–12, 15]. However, these De Bruijn sequences are obtained by a very little change of the base LFSRs, and stream ciphers based on these maximum-length FSRs may susceptible to algebraic attacks and correlation attacks.

Constructing De Bruijn sequences by joining the cycles of a nonlinear feedback shift register (NFSR) is a challenging work, because many fundamental problems related to NFSRs are essentially unsolved until now. Jansen et al. [8] proposed an algorithm for joining cycles of an arbitrary FSR. The efficiency of their algorithm depends on the length of the longest cycle in the base FSR. To find FSRs that contain only very short cycles, they turned to the LFSRs and constructed a class of such LFSRs. Their algorithm was improved recently in [13] where an improved version of the cycle joining algorithm was given and a large class of NFSRs that contain only very short cycles were proposed. Besides these universal algorithms, some special class of NFSRs were also analysed in order to constructing De Bruijn sequences. For example, the NFSRs with characteristic function $f * l$ were analysed in [20], where f is the characteristic function of a maximum length FSR and l is the characteristic function of a maximum length LFSR. Based on these NFSRs, approximately $O(2^{2^{k-1}+n})$ maximum length FSRs can be constructed, where k and n are the orders of f and l respectively. It was shown that, the time complexity to generate such a maximum length FSR is $O(k \cdot 2^{2k} + n(k+1)2^k + (2k+3) \cdot n^3)$.

In this paper, we consider the symmetric FSRs and construct a large class of De Bruijn sequences from them. Symmetric FSRs which is a special class of nonlinear FSRs were first studied in [9] where the cycle structure of some symmetric FSRs were determined. Then the research was continued in [16–19] where the general case was studied. It was proved that, the cycles in a symmetric FSR can be divided into layers according to the weights of states in cycles [16]. The cycles in a symmetric FSR can be joined together by using the general algorithms proposed in [8] and [13]. However, these algorithms provide us only one full cycle from a given FSR. In order to construct more full cycles, we select a special state from each layer of the base symmetric FSR, then the special states are used as bridging states in the process of cycle joining. Since different choices of the special states corresponding to different full cycles, by using this method, we can construct a large class of full cycles from just one symmetric FSR. In order to improve the efficiency of the algorithm, a special class of symmetric FSRs (we call them scattered symmetric FSRs) are used as the base FSR. It is shown that, at least $O(2^{\frac{n-6}{2}\log n})$ De Bruijn sequences of order n can be constructed from just one n-stage scattered symmetric FSR, and it requires no more than $2n$ comparisons and $n+1$ FSR shifts to generate the next state in the full cycle from the current state. By further analyse the cycle structure of the scattered symmetric FSRs, other methods for constructing De Bruijn sequences are suggested.

The paper is organized as follows. In Section 2, we introduce some necessary preliminaries. In Section 3, an algorithm for joining the cycles in a symmetric FSR is proposed. In Section 4, a special class of symmetric FSRs are analysed and other methods to join the cycles in these FSRs are suggested. In Section 5, we make a conclusion about our work.

2 Preliminaries

2.1 Symmetric Boolean functions

Let $\mathbb{F}_2 = \{0, 1\}$ be the finite field of two elements, and $\mathbb{F}_2^n$ be the vector space of dimension n over $\mathbb{F}_2$. For a vector $\mathbf{S} = (s_0, s_1, \ldots, s_{n-1})$, its weight is defined as the number of ones among the s_i's, i.e., $W(\mathbf{S}) = \sum_{i=0}^{n-1} s_i$. Sometimes, we regard $\mathbf{S}$ as an integer $\mathbf{S} = \sum_{i=0}^{n-1} s_i 2^{n-1-i}$. A Boolean function $f(x_0, x_1, \ldots, x_{n-1})$ in n variables is a mapping from $\mathbb{F}_2^n$ to $\mathbb{F}_2$. It is well known that it can be uniquely represented by its algebraic normal form (ANF), which is a multivariate polynomial. A symmetric Boolean function is a Boolean function whose value does not depend on the permutation of its input bits, i.e., it depends only on the number of ones in the input. Define the symmetric Boolean functions $E_k(x_1, x_2, \ldots, x_{n-1})$ for $k \in \{0, 1, \ldots, n-1\}$ by the equivalence: $E_k(s_1, s_2, \ldots, s_{n-1}) = 1$ if and only if $W(s_1, s_2, \ldots, s_{n-1}) = k$. Then it is easy to see, $\{E_k : k = 0, 1, \ldots, n-1\}$ is a basis for the vector space of all the symmetric Boolean functions in the variables $x_1, x_2, \ldots, x_{n-1}$ [16].

Lemma 1. *Let $h(x_1, x_2, \ldots, x_{n-1})$ be a symmetric Boolean function, then there exists an unique subset $M \subset \{0, 1, \ldots, n-1\}$ such that $h(x_1, x_2, \ldots, x_{n-1}) = \sum_{k \in M} E_k$.*

The subset M can be determined easily from the symmetric Boolean function h. To determine if $k \in M$ or not, we just need to test whether $h(0, \ldots, 0, \overbrace{1 \ldots, 1}^{k}) = 1$ or not. Therefore, by $n+1$ times test, the subset M is determined. For convenience, the subset M that determined by h is denoted by $\mathrm{Ind}(h)$.

2.2 Feedback shift registers

An n-stage feedback shift register (FSR) consists of n binary storage cells and a characteristic function f regulated by a single clock. In what follows, the characteristic function f is supposed to be nonsingular, i.e., of the form $f = x_0 + f_0(x_1, \ldots, x_{n-1}) + x_n$. The feedback function of this FSR is defined as $F(x_0, x_1, \ldots, x_{n-1}) = x_0 + f_0(x_1, \ldots, x_{n-1})$. The FSR with characteristic function f is denoted by $\mathrm{FSR}(f)$. At every clock pulse, the current state $(s_0, s_1, \ldots, s_{n-1})$ is updated by $(s_1, s_2, \ldots, s_{n-1}, F(s_0, s_1, \ldots, s_{n-1}))$. From an initial state $\mathbf{S}_0 = (s_0, s_1, \ldots, s_{n-1})$, after consecutive clock pulses, $\mathrm{FSR}(f)$ will generate a cycle $C = [\mathbf{S}_0, \mathbf{S}_1, \ldots, \mathbf{S}_{l-1}]$, where $\mathbf{S}_{i+1}$ is the next state of $\mathbf{S}_i$ for $i = 0, 1, \ldots, l-2$ and $\mathbf{S}_0$ is the next state of $\mathbf{S}_{l-1}$. The cycle C can also be denoted by $C = [s_0, s_1, \ldots, s_{l-1}]_n$ or simply $C = [s_0, s_1, \ldots, s_{l-1}]$, where s_i is the first component of $\mathbf{S}_i$ for $i = 0, 1, \ldots, l-1$. In this way, the set $\mathbb{F}_2^n$ is divided into cycles $C_1, C_2, \ldots, C_k$ by $\mathrm{FSR}(f)$, and vice versa, it is easy to see, a partition of $\mathbb{F}_2^n$ into cycles determines an n-stage FSR. So we can treat $\mathrm{FSR}(f)$ as a set of cycles and use the notation $\mathrm{FSR}(f) = \{C_1, C_2, \ldots, C_k\}$. The output sequences of $\mathrm{FSR}(f)$, denoted by $G(f)$, are the 2^n sequences $\mathbf{s} = s_0 s_1 \ldots$, such that $s_{t+n} = F(s_t, s_{t+1}, \ldots, s_{t+n-1})$ for $t \geq 0$. Since f is nonsingular, $G(f)$ contains only periodic sequences [5]. An FSR is called a linear feedback shift register (LFSR) if its characteristic function f is linear, i.e., f is of the form $f(x_0, x_1, \ldots, x_n) = a_0 x_0 + a_1 x_1 + \cdots + a_n x_n$, and nonlinear feedback shift register (NFSR) otherwise. For an n-stage FSR, the period of its output sequence is no more than 2^n. If this value is attained, we call the sequence De Bruijn sequence, and the FSR maximum-length FSR. There is only one cycle in a maximum-length FSR, and this cycle is usually called a full cycle or a De Bruijn cycle. We call FSR(f) a symmetric FSR if f is of the form $f = x_0+h(x_1, x_2, \ldots, x_{n-1})+x_n$ for some symmetric Boolean function h. The cycle structure of symmetric FSRs has been studied in [9, 16–19]. We recall some of the results in [16]. For simplicity, we use $[a, b]$, where a and b are two integers such that $a \le b$, to denote the integers lie between a and b, that is, $[a, b] = \{i \text{ is an integer } | a \le i \le b\}$.

Lemma 2. *[16] Let $h(x_1, x_2, \ldots, x_{n-1})$ be a symmetric Boolean function, and $\mathrm{Ind}(h) = \cup_{i=1}^{u}[a_i, b_i]$, where a_i and b_i are integers such that $b_i + 1 < a_{i+1}$. Define $\mathrm{Rem}(h) = \{r | 0 \le r \le n, a \notin \cup_{i=1}^{u}[a_i, b_i + 1]\}$. Let C be a cycle in $\mathrm{FSR}(x_0 + h + x_n)$, then there are two cases may happen:*

1. *There exists some integer $1 \le i \le u$ such that the weights of the states on C are lie between a_i and $b_i + 1$, that is, $a_i \le W(\mathbf{S}) \le b_i + 1$ for any $\mathbf{S} \in C$.*

2. *There exists some integer $r \in \mathrm{Rem}(h)$ such that the weights of the states on C are equal to r, that is, $W(\mathbf{S}) = r$ for any $\mathbf{S} \in C$.*

With the notations in Lemma 2, we define the layer of weight $[a_i, b_i+1]$ in $\mathrm{FSR}(x_0+h+x_n)$ to be $\mathcal{A}[a_i, b_i+1]_h = \{C | C \in \mathrm{FSR}(x_0+h+x_n), a_i \le W(\mathbf{S}) \le b_i+1 \text{ for any } \mathbf{S} \in C\}$ for $i = 1, 2, \ldots, u$, and the layer of weight $[r]$ in $\mathrm{FSR}(x_0 + h + x_n)$ to be $\mathcal{A}[r]_h = \{C | C \in \mathrm{FSR}(x_0 + h + x_n), W(\mathbf{S}) = r \text{ for any } \mathbf{S} \in C\}$ for $r \in \mathrm{Rem}(h)$. The subscript h is usually dropped if it is clear from the context which h is intended. Let $\mathrm{Rem}(h) = \{r_1, r_2, \ldots, r_v\}$ where v is the number of elements in $\mathrm{Rem}(h)$. Then according to Lemma 2, the cycles in $\mathrm{FSR}(x_0 + h + x_n)$ can be divided into $u + v$ layers: $\mathrm{FSR}(x_0 + h + x_n) = (\cup_{i=1}^{u}\mathcal{A}[a_i, b_i + 1]) \bigcup (\cup_{j=1}^{v}\mathcal{A}[r_j])$.

2.3 The cycle joining method

For a state $\mathbf{S} = (s_0, s_1, \ldots, s_{n-1})$, its companion is defined as $\widetilde{\mathbf{S}} = (s_0, s_1, \ldots, \overline{s}_{n-1})$, where $\overline{s}_{n-1}$ is the binary complement of s_{n-1}. Two cycles C_1 and C_2 are said to be adjacent if they are disjoint and there exists a state $\mathbf{S}$ on C_1 whose companion $\widetilde{\mathbf{S}}$ is on C_2. By interchanging the predecessors of $\mathbf{S}$ and $\widetilde{\mathbf{S}}$, the two cycles C_1 and C_2 are joined together. This is the basic idea of the cycle joining method introduced in [5]. Maximum length FSR can be obtained by joining the cycles in an FSR that producing several short cycles. For a given FSR, different ways to select the bridging states result in different full cycles. To count the number of full cycles obtained from a given FSR by the cycle joining method, we need the following definition.

Definition 1. *[6, 14] For an FSR, its adjacency graph is an undirected graph where the vertexes correspond to the cycles in it, and there exist m edges between two vertexes if and only if the two vertexes share m conjugate pairs. For simplicity, the m edges are usually replaced by an edge labeled with m.*

In graph theory, a spanning tree T of an undirected graph G is a connected subgraph that includes every vertex of G and contains no cycles. It is easy to see, there is an one-to-one correspondence between the spanning trees of the adjacency graph of FSR(f) and the full cycles generated from FSR(f) by the cycle joining menthod, because this represents a choice of adjacencies that repeatedly join two cycles into one ending with exactly one cycle, i.e., full cycle.

3 Joining the Cycles in a Symmetric FSR

Let $h(x_1, x_2, \ldots, x_{n-1})$ be a symmetric Boolean function. According to Lemma 2, the cycles in FSR(x_0+h+x_n) can be divided into $u+v$ layers: $\text{FSR}(x_0+h+x_n) = (\cup_{i=1}^{u}\mathcal{A}[a_i, b_i+1])\bigcup(\cup_{j=1}^{v}\mathcal{A}[r_j])$. For two layers $\mathcal{A}_1$ and $\mathcal{A}_2$ in FSR($x_0 + h + x_n$), we say $\mathcal{A}_1$ is lighter than $\mathcal{A}_2$ if the weights of the states in $\mathcal{A}_1$ is less than those in $\mathcal{A}_2$. For a cycle C in FSR($x_0 + h + x_n$) that does not contain the zero state $\mathbf{0}$, the cycle representative of C is defined as the numerically largest state $\mathbf{S}$ on C such that: $\mathbf{S}$ contains the longest run of ZEROS and is of the form $(*, \ldots, *, \overbrace{0, \ldots, 0}^{t}, 1)$, where t is the length of the longest run of ZEROS [13]. For the cycle that contains the zero state, there is no cycle representative. The cycles in FSR($x_0 + h + x_n$) can be joined together with the help of cycle representatives as shown in [13]. However, this method generate only one full cycle from a given FSR. In this section, we present an algorithm which provide us more full cycles from just one symmetric FSR.

At first, a special state is chosen from each layer. For each $1 \le i \le u$, we choose a state $\mathbf{S}[a_i]$ from the layer $\mathcal{A}[a_i, b_i + 1]$ such that: (1) $W(\mathbf{S}[a_i]) = a_i$, (2) $\mathbf{S}[a_i]$ is an odd state (regard states as integers), and (3) $\mathbf{S}[a_i]$ is not the representative of the cycle it belongs to. For each $1 \le j \le v$, we choose a state $\mathbf{S}[r_j]$ from the layer $\mathcal{A}[r_j]$ such that: (1) $\mathbf{S}[r_j]$ is an odd state, and (2) $\mathbf{S}[r_j]$ is not the representative of the cycle it belongs to. Define V be the set of these special states, that is, $V = \{\mathbf{S}[a_1], \ldots, \mathbf{S}[a_u], \mathbf{S}[r_1], \ldots, \mathbf{S}[r_v]\}$. We should note that, for some layers such a special state may not exists (in this case, no special state is chosen from this layer). Thus, the number of states in V is no more than $u + v$. However, in the case $2 \le a_i \le n-1$ (or $2 \le r_j \le n-1$) the special state in the layer $\mathcal{A}[a_i, b_i + 1]$ (or $\mathcal{A}[r_j]$) always exists.

Theorem 1. *In the case $2 \le a_i \le n-1$, the number of choices for $\mathbf{S}[a_i]$ in the layer $\mathcal{A}[a_i, b_i + 1]$ is no less than $\binom{n-2}{a_i-2}$. Similarly, in the case $2 \le r_j \le n-1$, the number of choices for $\mathbf{S}[r_j]$ in the layer $\mathcal{A}[r_j]$ is no less than $\binom{n-2}{r_j-2}$. For a given FSR($x_0 + h + x_n$) there are at least*

$$\prod_{2\le a_i\le n-1}\binom{n-2}{a_i-2} \cdot \prod_{2\le r_j\le n-1}\binom{n-2}{r_j-2}$$

choices for the set V.

Proof. It can be verified that, in the case $2 \le a_i \le n-1$, $\mathbf{S}[a_i]$ can be any state of weight a_i and be of the form $(*, \ldots, *, 1, 1)$. Therefore, $\mathbf{S}[a_i]$ has at least $\binom{n-2}{a_i-2}$ choices. Similarly, in the case $2 \le r_j \le n-1$, the number of choices for $\mathbf{S}[r_j]$ is no less than $\binom{n-2}{r_j-2}$. Therefore, for a given FSR($x_0 + h + x_n$), there are at least $\prod_{2\le a_i\le n-1}\binom{n-2}{a_i-2} \cdot \prod_{2\le r_j\le n-1}\binom{n-2}{r_j-2}$ choices for the set V. $\square$

Since the special states are chosen from different layers, for any cycle in FSR(x_0+h+x_n), there are at most one special state on this cycle. Let $C_0, C_1, \ldots, C_k$ be the cycles in FSR($x_0 + h + x_n$).

Without lose of generality, we assume C_0 is the cycle that contains the zero state. By the definition of cycle representative, C_0 contains no cycle representative. Let $\mathcal{A}_0$ be the layer in $\mathrm{FSR}(x_0+h+x_n)$ that contains the cycle C_0. By the definition of special state, it is easy to see that, there are no special state in $\mathcal{A}_0$, therefore, there are no special state on the cycle C_0. We can assume $C_1, C_2, \ldots, C_t$ are the cycles that each contains a special state, and $C_{t+1}, C_{t+2}, \ldots, C_k$ are the cycles that each does not contain a special state. For the cycles $C_1, C_2, \ldots, C_t$, their representatives are chosen to form a set R_1, that is, $R_1 = \{\text{the cycle representative of } C_i | i = 1, 2, \ldots, t\}$. For the cycles $C_{t+1}, C_{t+2}, \ldots, C_k$, their representatives are chosen to form a set R_2, that is, $R_2 = \{\text{the cycle representative of } C_i | i = t+1, t+2, \ldots, k\}$. According to the definition of V and R_2, for any cycle C in $\mathrm{FSR}(x_0 + h + x_n)$ that does no contain the zero state, there is one state $\mathbf{S}$ on C such that $\mathbf{S} \in V \cup R_2$. All the states in $V \cup R_2$ are odd states. The cardinality of $V \cup R_2$ is k, i.e., $|V \cup R_2| = k$.

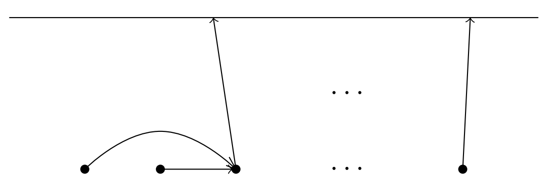
Theorem 2. *For a given* $\mathrm{FSR}(x_0 + h + x_n)$*. Let* V *and* R_2 *be the two sets defined above. If we interchange the predecessors of* $\mathbf{S}$ *and* $\widetilde{\mathbf{S}}$ *for every* $\mathbf{S} \in V \cup R_2$*, we get a full cycle.*

Proof. Let $C_0, C_1, \ldots, C_k$ be the cycles in $\mathrm{FSR}(x_0 + h + x_n)$, where C_0 is the cycle that contains the zero state, $C_1, C_2, \ldots, C_t$ are the cycles that each contain a special state and $C_{t+1}, C_{t+2}, \ldots, C_k$ are the cycles that each does not contain a special state. Let $\mathbf{S}_i$ be the special state on C_i for $i = 1, 2, \ldots, t$, and $\mathbf{S}_j$ be the cycle representative of C_j for $j = t+1, t+2, \ldots, k$. Let T be the directed graph that take $C_0, C_1, \ldots, C_k$ as his nodes, and there is a directed edge from C_i to C_j if and only if $\widetilde{\mathbf{S}}_i$ is on C_j. We need to show that T is a directed tree with root C_0.

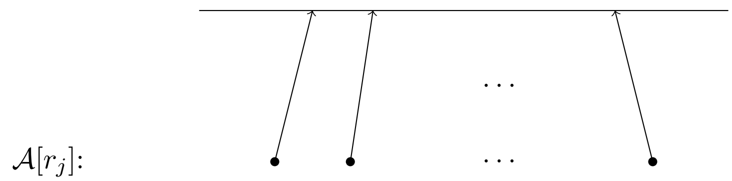
Suppose there is a directed edge from C_i to C_j. Since $\mathbf{S}_i$ is an odd state, we have $W(\widetilde{\mathbf{S}}_i) = W(\mathbf{S}_i) - 1$. If C_i and C_j belong to the same layer, then $\mathbf{S}_i$ is the cycle representative of C_i. Therefore, the length of the longest run of ZEROS in C_j is larger than the length of the longest run of ZEROS in C_i. If C_i and C_j lies on two different layers, then there are two cases may happen: (1) $\mathbf{S}_i$ is the cycle representative of C_i, or (2) $\mathbf{S}_i$ is a special state. In either case, the layer that contains C_j is lighter than the layer contains C_i. Therefore, there are no cycles in T. Considering also that there are k edges in T, we know T is a directed tree with root C_0. $\square$

The properties of the tree T defined in the proof of Theorem 2 are illustrated by the following two pictures. The cycles in $\mathrm{FSR}(x_0 + h + x_n)$ are divided into $u + v$ layers: $\mathrm{FSR}(x_0 + h + x_n) = (\cup_{i=1}^{u} \mathcal{A}[a_i, b_i + 1]) \bigcup (\cup_{j=1}^{v} \mathcal{A}[r_j])$. For a cycle C in the layer $\mathcal{A}[a_i, b_i + 1]$, the directed edge start from C will end at some cycle in the same layer or in the layer that lighter than $\mathcal{A}[a_i, b_i + 1]$.

$\mathcal{A}[a_i, b_i + 1]$:



For a cycle C in the layer $\mathcal{A}[r_j]$, the directed edge started from C will end at some cycle in the layer that lighter than $\mathcal{A}[r_j]$.



Based on Theorem 2, an algorithm for generating full cycles from $\text{FSR}(x_0+h+x_n)$ is presented. Given $\text{FSR}(x_0+h+x_n)$ and an initial state, the algorithm will generate a full cycle. This algorithm complements the value of the feedback function only if the odd successor $(s_{i+1}, \ldots, s_{i+n-1}, 1)$ of the current state $\mathbf{S} = (s_i, s_{i+1}, \ldots, s_{i+n-1})$ belongs to the set $V \cup R_2$.

Algorithm 1 Generation of full cycles based on a symmetric FSR

Input: A symmetric Boolean function h, an initial state $\mathbf{S}_0 = (s_0, s_1, \ldots, s_{n-1})$.

Output: A De Bruijn cycle $[\mathbf{S}_0, \mathbf{S}_1, \ldots, \mathbf{S}_{2^n-1}]$.

1: Choose and store the set V.
2: Determine and store the set R_1.
3: **for** $i \in \{0, 1, \ldots, 2^n - 1\}$ **do**
4:     Define $\mathbf{S} = (s_{i+1}, \ldots, s_{i+n-1}, 1)$.
5:     **if** $\mathbf{S} \in V$ **then**
6:         $\mathbf{S}_{i+1} = (s_{i+1}, \ldots, s_{i+n-1}, s_i + h(s_{i+1}, s_{i+2}, \ldots, s_{i+n-1}) + 1)$
7:     **else if** $\mathbf{S}$ is not a cycle representative **then**
8:         $\mathbf{S}_{i+1} = (s_{i+1}, \ldots, s_{i+n-1}, s_i + h(s_{i+1}, s_{i+2}, \ldots, s_{i+n-1}))$
9:     **else if** $\mathbf{S} \in R_1$ **then**
10:         $\mathbf{S}_{i+1} = (s_{i+1}, \ldots, s_{i+n-1}, s_i + h(s_{i+1}, s_{i+2}, \ldots, s_{i+n-1}))$
11:     **else**
12:         $\mathbf{S}_{i+1} = (s_{i+1}, \ldots, s_{i+n-1}, s_i + h(s_{i+1}, s_{i+2}, \ldots, s_{i+n-1}) + 1)$
13:     **end if**
14: **end for**

The cardinalities of the two sets V and R_1 are no more than n, that is, $|V| = |R_1| \leq n$. According to the proof of Theorem 1, the set V can be chosen in time $O(n^2)$. For each special state in V, traversing the cycle that contains this special state can determine the representative of this cycle. Therefore, the set R_1 can be determined at the cost of $n \cdot l$ FSR shifts, where l is the length of the longest cycle in $\text{FSR}(x_0 + h + x_n)$. We are more interested in the complexity of each step of the for-loop in Algorithm 1 because most times we only generate a tiny fraction of the full cycle. The most consuming part of the for-loop lies in line 5, line 7 and line 9. In the line 5, it needs at most n comparisons to test whether $\mathbf{S} \in V$ or not. Similarly, it needs at most n comparisons to test whether $\mathbf{S} \in R_1$ or not. In the line 7, by traversing the cycle that contains $\mathbf{S}$, we can determine whether $\mathbf{S}$ is a cycle representative or not at the cost of no more than l FSR shifts. Therefore, we have the following theorem.

Theorem 3. *For a given* $\text{FSR}(x_0 + h + x_n)$*, Algorithm 1 can generate the next state from the* *current state at the cost of $2n$ comparisons and l FSR-shifts, where l is the length of the longest cycle in* $\mathrm{FSR}(x_0 + h + x_n)$.

This algorithm are not very efficient if a general symmetric FSR is used because the length of the longest cycle in this FSR may be very large. In the following, we will show that, for some special symmetric FSRs, this algorithm can be very fast. For a set M of integers, we say M is scattered if for any $k \in M$ we have $k - 1 \notin M$. We define the symmetric FSR with characteristic function $x_0 + h + x_n$ such that $\mathrm{Ind}(h)$ is scattered as scattered symmetric FSR. Some properties about the scattered symmetric FSRs are given in [16].

Lemma 3. *[16] Let* $\mathrm{FSR}(x_0 + h + x_n)$ *be an n-stage scatted symmetric FSR, and C be a cycle in* $\mathrm{FSR}(x_0 + h + x_n)$*. Then the length of C is a divisor of n or $n + 1$.*

The exact number of the n-stage scattered symmetric FSRs is not known to us, however, an obvious lower bound is given by $2^{\frac{n+1}{2}}$, since $\mathrm{Ind}(h)$ can be any subset of $\{\text{i is odd} | 0 \le i \le n-1\}$ or $\{\text{i is even} | 0 \le i \le n-1\}$. A scattered symmetric FSR contains only cycles of length no more than $n + 1$, therefore, Algorithm 1 is very fast if a scattered symmetric FSR is used as the base FSR.

Theorem 4. *Let* $\mathrm{FSR}(x_0 + h + x_n)$ *be an n-stage scattered symmetric FSR. Algorithm 1 can generate at least $O(2^{\frac{n-6}{2}\log n})$ De Bruijn sequences based on* $\mathrm{FSR}(x_0+h+x_n)$*. To generate the next state in the full cycle, it needs no more than $2n$ comparisons and $n+1$ FSR shifts.*

Proof. According to Theorem 1, there are at least $\prod_{2\le a_i\le n-1}\binom{n-2}{a_i-2}\cdot\prod_{2\le r_j\le n-1}\binom{n-2}{r_j-2}$ choices for the set V. Since $\mathrm{FSR}(x_0 + h + x_n)$ is scattered, for each even number $2 \le i \le n-2$, at least one of $\binom{n-2}{i-2}$ and $\binom{n-2}{i-1}$ lies in the production $\prod_{2\le a_i\le n-1}\binom{n-2}{a_i-2}\cdot\prod_{2\le r_j\le n-1}\binom{n-2}{r_j-2}$. Therefore, we have

$$\prod_{2\le a_i\le n-1}\binom{n-2}{a_i-2}\cdot\prod_{2\le r_j\le n-1}\binom{n-2}{r_j-2} \ge \prod_{\substack{2\le i\le n-2,\\ i \text{ is even}}} min\left\{\binom{n-2}{i-2},\binom{n-2}{i-1}\right\}.$$

Then the first assertion of this theorem can be proved as follows,

$$\begin{aligned}\prod_{\substack{2\le i\le n-2,\\ i \text{ is even}}} min\left\{\binom{n-2}{i-2},\binom{n-2}{i-1}\right\} &= \prod_{\substack{4\le i\le n-2,\\ i \text{ is even}}} min\left\{\binom{n-2}{i-2},\binom{n-2}{i-1}\right\}\\ &\ge \prod_{\substack{4\le i\le n-2,\\ i \text{ is even}}}(n-2) = (n-2)^{\frac{n-6}{2}} = O\left(n^{\frac{n-6}{2}}\right) = O(2^{\frac{n-6}{2}\log n}).\end{aligned}$$

For the second assertion, it can be verified easily according to Theorem 3 and Lemma 3. ☐

4 Other Methods for Joining Cycles

The scattered symmetric FSR is further studied in this section. Some properties of these FSRs are given, and other methods for joining the cycles in these FSRs are suggested.

Let $\mathrm{FSR}(x_0 + h + x_n)$ be a scattered symmetric FSR. Define P to be the set of odd integers in $\mathrm{Ind}(h)$, Q to be the set of even integers in $\mathrm{Ind}(h)$, and $\mathrm{Rem}(h) = \{0 \le i \le n | i \notin \mathrm{Ind}(h), i-1 \notin$

$\mathrm{Ind}(h)\}$. Then according to Lemma 2, the cycles in $\mathrm{FSR}(x_0 + h + x_n)$ can be divided into layers, $\mathrm{FSR}(x_0+h+x_n) = \big(\cup_{r\in \mathrm{Rem}(h)}\mathcal{A}[r]\big) \cup (\cup_{p\in P}\mathcal{A}[p, p+1]) \cup (\cup_{q\in Q}\mathcal{A}[q, q+1])$. Some properties about these layers are given in the following theorem.

Theorem 5. *With the notations above, we have $\mathcal{A}[r] \subset \mathrm{FSR}(x_0 + x_n)$, $\mathcal{A}[p, p+1] \subset \mathrm{FSR}(x_0 + x_1 + \cdots + x_n)$ and $\mathcal{A}[q, q+1] \subset \mathrm{FSR}(x_0 + x_1 + \cdots + x_n + 1)$ for any $r \in \mathrm{Rem}(h)$, $p \in P$ and $q \in Q$.*

Proof. Let C be a cycle in $\mathrm{FSR}(x_0 + h + x_n)$ and $\mathbf{S}$ be a state on C. We need to show that: (1) if $W(\mathbf{S}) \in \mathrm{Rem}(h)$, then $C \in \mathrm{FSR}(x_0 + x_n)$; (2) if $W(\mathbf{S}) \in P$ or $W(\mathbf{S}) - 1 \in P$, then $C \in \mathrm{FSR}(x_0+x_1+\cdots+x_n)$; and (3) if $W(\mathbf{S}) \in Q$ or $W(\mathbf{S})-1 \in Q$, then $C \in \mathrm{FSR}(x_0+x_1+\cdots+x_n+1)$. Denote $\mathbf{S}$ by $\mathbf{S} = (s_0, s_1, \ldots, s_{n-1})$. Let s_n be the next bit generated by $\mathrm{FSR}(x_0 + h + x_n)$ on the state $\mathbf{S}$, i.e., $s_n = s_0 + h(s_1, s_2, \ldots, s_{n-1})$.

Suppose $W(\mathbf{S}) \in \mathrm{Rem}(h)$. In the case of $s_0 = 0$, we have $W(s_1, s_2, \ldots, s_{n-1}) \in \mathrm{Rem}(h)$ which implies $W(s_1, s_2, \ldots, s_{n-1}) \notin \mathrm{Ind}(h)$. Therefore, we have $h(s_1, s_2, \ldots, s_{n-1}) = 0$ and $s_n = s_0$. In the case of $s_0 = 1$, we have $W(s_1, s_2, \ldots, s_{n-1}) - 1 \in \mathrm{Rem}(h)$ which also implies $W(s_1, s_2, \ldots, s_{n-1}) \notin \mathrm{Ind}(h)$. Therefore, we have $h(s_1, s_2, \ldots, s_{n-1}) = 0$ and $s_n = s_0$. Thus, we know that C is a cycle in $\mathrm{FSR}(x_0 + x_n)$.

Suppose $W(\mathbf{S}) \in P$ or $W(\mathbf{S}) - 1 \in P$. There are four cases need to be considered. In the case of $W(\mathbf{S}) \in P$ and $s_0 = 0$, we have $W(s_1, s_2, \ldots, s_{n-1}) \in \mathrm{Ind}(h)$ and $s_n = s_0 + h(s_1, s_2, \ldots, s_{n-1}) = 1$. Since $W(\mathbf{S})$ is odd, we have $s_0 + s_1 + \ldots + s_n = 0$. Similarly, for the cases of $W(\mathbf{S}) \in P$ and $s_0 = 1$, $W(\mathbf{S})-1 \in P$ and $s_0 = 0$, and $W(\mathbf{S})-1 \in P$ and $s_0 = 1$, we can also prove that $s_0+s_1+\ldots+s_n = 0$. Therefore, C is a cycle in $\mathrm{FSR}(x_0 + x_1 + \cdots + x_n)$

Suppose $W(\mathbf{S}) \in Q$ or $W(\mathbf{S}) - 1 \in Q$. There are four cases need to be considered. In the case of $W(\mathbf{S}) \in Q$ and $s_0 = 0$, we have $W(s_1, s_2, \ldots, s_{n-1}) \in \mathrm{Ind}(h)$ and $s_n = s_0 + h(s_1, s_2, \ldots, s_{n-1}) = 1$. Since $W(\mathbf{S})$ is even, we have $s_0 + s_1 + \ldots + s_n + 1 = 0$. Similarly, for the cases of $W(\mathbf{S}) \in Q$ and $s_0 = 1$, $W(\mathbf{S}) - 1 \in Q$ and $s_0 = 0$, and $W(\mathbf{S}) - 1 \in Q$ and $s_0 = 1$, we can also prove that $s_0 + s_1 + \ldots + s_n + 1 = 0$. Therefore, C is a cycle in $\mathrm{FSR}(x_0 + x_1 + \cdots + x_n + 1)$ □

Example 1. *Let $h(x_1, x_2, \ldots, x_4) = E_1 + E_4$ be a symmetric function. Then we have $P = \{1\}$, $Q = \{4\}$, and $\mathrm{Rem}(h) = \{0, 3\}$. The cycles in $\mathrm{FSR}(x_0 + h + x_5)$ can be divided into 4 layers, $\mathcal{A}[0]$, $\mathcal{A}[1, 2]$, $\mathcal{A}[3]$, and $\mathcal{A}[4, 5]$. These layers are shown in the following table.*

Table 1: The layers and cycles in $\mathrm{FSR}(x_0 + E_1 + E_4 + x_5)$

layers	cycles	contained in
$\mathcal{A}[0]$	$C_0 = [00000]$	$\subset \mathrm{FSR}(x_0 + x_5)$
$\mathcal{A}[1, 2]$	$C_1 = [00001, 00011, 00110, 01100, 11000, 10000]$ $C_2 = [00010, 00101, 01010, 10100, 01000, 10001]$ $C_3 = [00100, 01001, 10010]$	$\subset \mathrm{FSR}(x_0 + x_1 + \cdots + x_5)$
$\mathcal{A}[3]$	$C_4 = [00111, 01110, 11100, 11001, 10011]$ $C_5 = [01011, 10110, 01101, 11010, 10101]$	$\subset \mathrm{FSR}(x_0 + x_5)$
$\mathcal{A}[4, 5]$	$C_6 = [01111, 11111, 11110, 11101, 11011, 10111]$	$\subset \mathrm{FSR}(x_0 + x_1 + \cdots + x_5 + 1)$

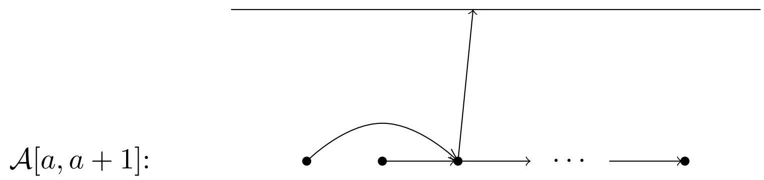
Etzion et al. [2] proposed two algorithms for joining the cycles in $\mathrm{FSR}(x_0 + x_n)$ and $\mathrm{FSR}(x_0 + x_1 + \cdots + x_n)$ respectively. The cycles in $\mathrm{FSR}(x_0 + x_1 + \cdots + x_n + 1)$ can be joined together in a similar way as that of $\mathrm{FSR}(x_0 + x_1 + \cdots + x_n)$ as noted in [2]. According to Theorem 5, the cycles in a scattered symmetric FSR are a selected combination of the cycles in $\mathrm{FSR}(x_0 + x_n)$, $\mathrm{FSR}(x_0 + x_1 + \cdots + x_n)$ and $\mathrm{FSR}(x_0 + x_1 + \cdots + x_n + 1)$. Therefore, by making small changes, the algorithms in [2] can be applied to scattered symmetric FSRs. Actually, there are many ways that can efficiently join the cycles in a scattered symmetric FSR. Different ways to choose the special states and different ways to define the representatives of cycles will result in different methods to joining cycles. As an illustration, we suggest one method in the following.

Let $\mathrm{FSR}(x_0 + h + x_n)$ be a scattered symmetric FSR. For a cycle C in $\mathrm{FSR}(x_0 + h + x_n) \cap \mathrm{FSR}(x_0 + x_n)$, the cycle representative of C is defined as in [13], that is, if C contains the zero state, then there is no representative on C, otherwise, the cycle representative of C is defined as the as the numerically largest state $\mathbf{S}$ on C such that: $\mathbf{S}$ contains the longest run of ZEROS and is of the form $(*, \ldots, *, \overbrace{0, \ldots, 0}^{t}, 1)$, where t is the length of the longest run of ZEROS. For a cycle C in $\mathrm{FSR}(x_0 + h + x_n) \cap \mathrm{FSR}(x_0 + x_1 + \cdots + x_n)$, the cycle representative of C is defined as in [2], that is, if C is a run-cycle, i.e., of the form $[1, 1, \ldots, 1, 0, 0, \ldots, 0]_n$, then there is no representative on C, otherwise, the cycle representative of C is defined as the numerically largest state $\mathbf{S}$ on C such that: $W(\mathbf{S})$ is even and $\mathbf{S}$ contains the longest run of ONES and is of the form $(\overbrace{0, \ldots, 0}^{r}, \overbrace{1, \ldots, 1}^{t}, *, \ldots, *, 1)$, where $r \geq 0$ and t is the length of the longest run of ONES. Similarly, for a cycle C in $\mathrm{FSR}(x_0 + h + x_n) \cap \mathrm{FSR}(x_0 + x_1 + \cdots + x_n + 1)$, if C is a run-cycle, then there is no representative on C, otherwise, the cycle representative of C is defined as the numerically largest state $\mathbf{S}$ on C such that: $W(\mathbf{S})$ is odd and $\mathbf{S}$ contains the longest run of ONES and is of the form $(\overbrace{0, \ldots, 0}^{r}, \overbrace{1, \ldots, 1}^{t}, *, \ldots, *, 1)$, where $r \geq 0$ and t is the length of the longest run of ONES

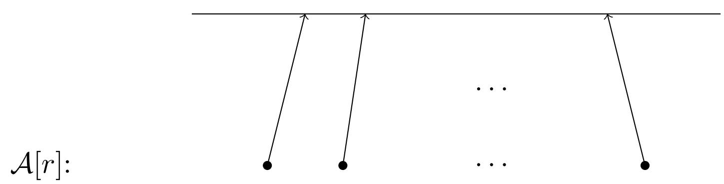
For each $a \in \mathrm{Ind}(h)$, we choose a state $\mathbf{S}[a]$ from the layer $\mathcal{A}[a, a+1]$ such that: (1) $W(\mathbf{S}[a]) = a$, and (2) $\mathbf{S}[a]$ is an odd state. Define V be the set of these special states, that is, $V = \{\mathbf{S}[a] | a \in \mathrm{Ind}(h)\}$. We should note that, in the case of $a \neq 0$ the special state in the layer $\mathcal{A}[a, a+1]$ always exists. Let R be the set of cycle representatives, that is, $R = \{$the cycle representative of $C | C \in \mathrm{FSR}(x_0 + h + x_n)\}$. Then we have the following theorem. The proof of this theorem is quite similar to that of Theorem 2 and so is omitted.

Theorem 6. *Let $\mathrm{FSR}(x_0 + h + x_n)$ be a scattered symmetric FSR. Let V and R be the two sets defined above. If we interchange the predecessors of $\mathbf{S}$ and $\widetilde{\mathbf{S}}$ for every $\mathbf{S} \in V \cup R$, we get a full cycle.*

Let $C_0, C_1, \ldots, C_k$ be the cycles in $\mathrm{FSR}(x_0 + h + x_n)$. Let T be the directed graph that take $C_0, C_1, \ldots, C_k$ as his nodes, and there is a directed edge from C_i to C_j if and only if there is a special state S on C_i whose companion $\widetilde{\mathbf{S}}_i$ is on C_j or the companion of the representative of C_i lies on C_j, then T is a tree. The properties of the tree T are shown by the following two pictures. Let C be a cycle in the layer $\mathcal{A}[a, a+1]$. In the case of C does not contain a special state, the directed edge start from C (if it exists) will end at some cycle in the same layer. In the case of C contains a special state, the directed edges start from C (one or two) will end at some cycle in the same layer or some cycle in the layer that lighter than $\mathcal{A}[a, a+1]$.

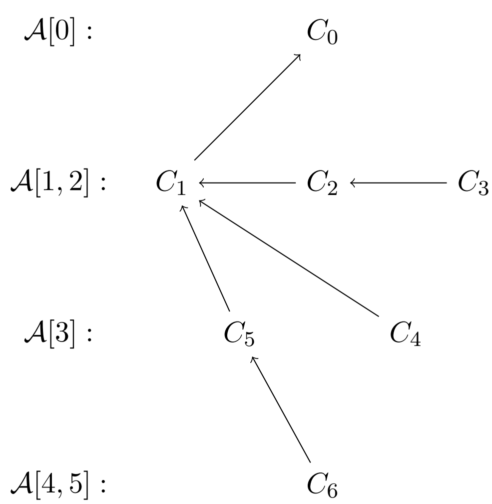


For a cycle C in the layer $\mathcal{A}[r]$, the directed edge started from C will end at some cycle in the layer that lighter than $\mathcal{A}[r]$.



An example is given to illustrate the process of the cycle joining algorithm proposed in this section.

Example 2. *The scattered symmetric FSR in Example 1 is used as the base FSR. The set* V *can be chosen as* $V = \{(00001), (10111)\}$. *By the definition,* $R = \{(10001), (01001), (11001), (01101)\}$. *The adjacency tree is shown below.*



The resulting De Bruijn sequence is: [00000110101111101100111000101001]

5 Conclusion

The symmetric FSRs are used to construct De Bruijn sequences in this paper. From an n-stage scattered symmetric FSR, at least $O(2^{\frac{n-6}{2}\log n})$ De Bruijn sequences of order n are constructed. To generate the next bit in the De Bruijn sequence from the current state, it requires no more than $2n$ comparisons and $n+1$ FSR shifts. Some properties of the cycle structure of scattered symmetric FSRs are given, and by these properties other ways to join cycles are suggested.

References

[1] N. G. de Bruijn, “A combinatorial problem,” Proc. Kon. Ned. Akad. Wetensch, vol. 49, pp. 758-746, 1946.

[2] T. Etzion and A. Lempel, “Algorithms for the generation of full-length shift-register sequences,” IEEE Trans. Inf. Theory, vol. 30, no. 3, pp. 480-484, 1984.

[3] H. Fredricksen, “A class of nonlinear de Bruijn cycles,” J. Comb. Theory, Ser. A, vol. 19, no. 2, pp. 192-199, 1975.

[4] H. Fredricksen, “A survey of full length nonlinear shift register cycle algorithms,” SIAM Rev., vol.24, no. 2, pp. 195-221, 1982.

[5] S. W. Golomb, Shift Register Sequences, San Francisco, CA: Holden-Day, 1967.

[6] E. R. Hauge and J. Mykkeltveit, “On the classification of deBruijn sequences,” Discrete Math., vol. 148, no. 1, pp. 65-83, 1996.

[7] F. Hemmati, “A large class of nonlinear shift register sequences,” IEEE Trans. Inf. Theory, vol. 28, no. 2, pp. 355-359, 1982.

[8] C. J. A. Jansen, W. G. Franx and D. E. Boekee, “An efficient algorithm for the generation of deBruijn cycles,” IEEE Trans. Inf. Theory, vol. 37, no. 5, pp. 1475-1478, 1991.

[9] K. Kjeldsen, “On the cycle structure of a set of nonlinear shift registers with symmetric feedback functions,” J. Comb. Theory, Ser. A, vol. 20, no. 2, pp. 154-169, 1976.

[10] C.Y. Li, X.Y. Zeng, T. Helleseth, C.L. Li and L. Hu, “The properties of a class of linear FSRs and their applications to the construction of nonlinear FSRs,” IEEE Trans. Inf. Theory, vol. 60, no. 5, pp. 3052-3061, 2014.

[11] C.Y. Li, X.Y. Zeng, C.L. Li, and T. Helleseth, “A Class of De Bruijn Sequences,” IEEE Trans. Inf. Theory, vol. 60, no. 12, pp. 7955-7969, 2014.

[12] M. Li, Y.P. Jiang, D.D. Lin, “The adjacency graphs of some feedback shift registers,” in Cryptology ePrint Archive [online]. Available: http://eprint.iacr.org/2014/658, 2014.

[13] M. Li, D.D. Lin, “De Bruijn sequences from nonlinear feedback shift registers,” in Cryptology ePrint Archive [online]. Available: http://eprint.iacr.org/2015/667, 2015.

[14] K. B. Magleby, “The synthesis of nonlinear feedback shift registers,” Tech. Rep. 6207-1, Stanford Electronic Labs, Stanford, CA, 1963.

[15] J. Mykkeltveit, M. K. Siu and P. Tong, “On the cycle structure of some nonlinear shift register sequences,” Inf. Contr., vol. 43, no. 2, pp. 202-215, 1979.

[16] Jan Sϕreng, “The periods of the sequences generated by some symmetric shift registers,” J. Comb. Theory, Ser. A, vol. 21, no. 2, pp. 164-187, 1976.

[17] Jan Sϕreng, “Symmetric shift registers,” Pacific J. Math., vol. 85, no. 1, pp. 201-229, 1976.

[18] J. Sϕreng, “Symmetric shift registers part 2,” Pacific J. Math., vol. 98, no. 1, pp. 203-234, 1982.

[19] Z.X. Wang, H. Xu, W.F. Qi, “On the cycle structure of some nonlinear feedback shift registers,” Chinese Journal of Electronics, vol. 23, no. 4, pp. 801-804, 2014.

[20] X.X. Zhao, W.F. Qi, “The construction of de Bruijn sequences based on cascade connection,” Journal of Crypologic Research (in Chinese), vol. 2, no. 3, pp. 245-257, 2015.