# De Bruijn Sequences from Joining Cycles of Nonlinear Feedback Shift Registers

Ming Li[1], Cees J.A. Jansen[2], Dongdai Lin[1], and Qiuyan Wang[3]

**Abstract**

De Bruijn sequences are a class of nonlinear recurring sequences that have wide applications in cryptography and modern communication systems. One main method for constructing them is to join the cycles of a feedback shift register (FSR) into a full cycle, which is called the cycle joining method. Jansen et al. (IEEE Trans on Information Theory 1991) proposed an algorithm for joining cycles of an arbitrary FSR. This classical algorithm is further studied in this paper. Motivated by their work, we propose a new algorithm for joining cycles, which doubles the efficiency of the classical cycle joining algorithm. Since both algorithms need FSRs that only generate short cycles, we also propose efficient ways to construct short-cycle FSRs. These FSRs are nonlinear and are easy to obtain. As a result, a large number of de Bruijn sequences are constructed from them. We explicitly determine the size of these de Bruijn sequences. Besides, we show a property of the pure circulating register, which is important for searching for short-cycle FSRs.

**Keywords**: De Bruijn sequence, feedback shift register, cycle joining algorithm.

## 1 Introduction

Binary de Bruijn sequences of order $n$ are sequences of period $2^n$ such that each $n$-tuple appears exactly once in one period. These sequences have many favorable properties, such as long period, large linear span and good randomness, and they are widely used in cryptography and modern communication systems [7]. It is well known [3] that the number of $n$-th order de Bruijn sequences is $2^{2^{n-1}-n}$. Even though the number of sequences of given order is very large, today it is only known how to efficiently construct small fractions of this large number [4, 10, 20, 21, 27]. An excellent survey of algorithms for generating de Bruijn sequences is given in [7].

A long-standing method for constructing de Bruijn sequences is to consider a feedback shift register (FSR) producing several cycles which are joined together to form a full cycle. This method is known as the cycle joining method, proposed by Golomb [8]. Linear feedback shift registers (LFSRs) with simple cycle structures are often used with this method. Fredricksen

[1]M. Li and D.D. Lin are with the State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China. E-mail: {liming,ddlin}@iie.ac.cn.

[2]C.J.A. Jansen is a former professor of the Eindhoven University of Technology (TU/e), Eindhoven, the Netherlands, and visiting professor at the University of Bergen (UiB), Bergen, Norway. E-mail: info@ceesjansen.nl.

[3]Q.Y. Wang is with the School of Computer Science and Software Engineering, Tianjin Polytechnic University, Tianjin, China. E-mail: wangqiuyan@tjpu.edu.cn.

[6], Etzion and Lempel [5] showed how to generate de Bruijn sequences from pure circulating registers and pure summing registers, respectively. Recent results include the use of LFSRs with characteristic polynomials of the forms $(1+x)^m p(x)$ and $(1+x^m)p(x)$, where $p(x)$ is a primitive polynomial and $m$ is a positive integer $\leq 5$, to generate de Bruijn sequences [11, 14–18, 24]. By joining the cycles of an LFSR, a large class of de Bruijn sequences can be constructed efficiently. Although progress has been made in joining cycles of LFSRs, the problem of joining cycles of a general LFSR with arbitrary feedback function is still unsolved, because of the difficulty of the so called adjacency relations between the cycles of such a general LFSR.

Compared with the linear case, there is little progress in joining cycles of nonlinear feedback shift registers (NFSRs), especially in an efficient way. From a viewpoint of cryptography, de Bruijn sequences based on NFSRs are more preferable than those based on LFSRs. This is due to the many long linear subsequences in these de Bruijn sequences, by which the stream ciphers based on them may be susceptible to some classic attacks [2, 28].

Jansen et al. [12] proposed an algorithm for joining cycles of an arbitrary FSR. For a given FSR, they define the numerically least state (regard a state as an integer) on each cycle as its representative. Then they showed that, interchanging the predecessors of the cycle representatives with those of their companions results in a full cycle. For the application of this algorithm, one needs to test whether a state is the cycle representative or not at every step. Therefore, the performance of the algorithm depends on the length of the longest cycle in the FSR. Consequently, FSRs producing only short cycles are needed. To find such FSRs, they resorted to LFSRs. By multiplying a large number of irreducible polynomials of the same degree, a polynomial with low period (relative to its degree) is obtained. LFSRs taking such polynomials as their characteristic polynomials generate only short cycles, and they can be used to construct de Bruijn sequences efficiently.

Motivated by the research of Jansen et al. [12], we propose a new algorithm for joining cycles of an arbitrary FSR, by using a new definition of cycle representative. Compared to the work in [12], the new cycle representatives double the efficiency of the cycle joining algorithm. Furthermore, the choices of the new cycle representatives are more flexible, that is, there are more than one ways to choose a representative of a given cycle. To find FSRs producing only short cycles, we resort to NFSRs. We propose two classes of NFSRs with only short cycles that are easy to obtain. The first class is constructed by using the operation $*$ (see Section 4) and the second class originates from symmetric FSRs (see Section 5). The number of de Bruijn sequences constructed in both classes is also considered. Besides, we show that, the $n$-stage pure circulating register is the unique $n$-stage FSR that generate cycles of length no more than $n$ (see Section 6).

The remainder of this paper is organized as follows. Section 2 introduce some preliminaries. In Section 3, a new algorithm for joining cycles of arbitrary FSRs is presented. Section 4 proposes a way to obtain short cycle NFSRs. Section 5 considers symmetric shift registers, and some other methods to construct de Bruijn sequences are also discussed. In Section 6 we show a property of the pure circulating register. Section 7 gives the final conclusions of this paper.

## 2 Preliminaries

### 2.1 Boolean Functions

Let $\mathbb{F}_2 = \{0, 1\}$ be the binary Galois field, and $\mathbb{F}_2^n$ be the $n$th-dimensional vector space over $\mathbb{F}_2$. An $n$-variable Boolean function $f(x_0, x_1, \ldots, x_{n-1})$ is a function from $\mathbb{F}_2^n$ to $\mathbb{F}_2$. It is well known that it can be uniquely written in the so-called algebraic normal form (ANF) as follows:

$$f(x_0, x_1, \ldots, x_{n-1}) = \sum_{I \in \mathcal{P}(\mathbb{N})} a_I \cdot \prod_{i \in I} x_i,$$

where $a_I \in \mathbb{F}_2$ and $\mathcal{P}(\mathbb{N})$ is the power set of $\{0, 1, \ldots, n-1\}$. The degree of $f$, denoted by $\deg f$, is defined to be $\deg f = \max\{|I| \mid a_I \neq 0\}$. If $\deg f \geq 1$, then the highest subscript $i$ of $x_i$ that occurs in the ANF of $f$ is called the order of $f$ and denoted by $\mathrm{ord}(f)$. For example, for the Boolean function $f = x_0 + x_1 + x_1x_3 + x_4$ we have $\deg f = 2$ and $\mathrm{ord}(f) = 4$. For two Boolean functions $f(x_0, x_1, \ldots, x_n)$ and $g(x_0, x_1, \ldots, x_m)$, we denote

$$f * g = f(g(x_0, x_1, \ldots, x_m), g(x_1, x_2, \ldots, x_{m+1}), \ldots, g(x_n, x_{n+1}, \ldots, x_{n+m})),$$

which is a Boolean function in $n + m + 1$ variables. It is easy to see that $\mathrm{ord}(f * g) = \mathrm{ord}(f) + \mathrm{ord}(g)$. The operation $*$ is not commutative, that is, $f * g$ and $g * f$ are not the same in general. We refer the reader to [9, 24] for more details about this operation. The Hamming weight of a vector $\mathbf{S} = (s_0, s_1, \ldots, s_{n-1})$, denoted by $w_H(\mathbf{S})$, is the number of 1's in $\mathbf{S}$. The Hamming weight of a Boolean function $f(x_0, x_1, \ldots, x_{n-1})$, denoted by $w_H(f)$, is the size of its support $\{x \in \mathbb{F}_2^n \mid f(x) \neq 0\}$. For two Boolean functions $f$ and $g$ of the same order, their distance $d_H(f, g)$ is defined to be the size of the set $\{x \in \mathbb{F}_2^n \mid f(x) \neq g(x)\}$. The following result about the distance between two Boolean functions comes from the minimal distance of the Reed-Muller codes, which are a family of linear error-correcting codes used in communications. We refer the reader to [23, 26] for the proof of this result.

**Lemma 1.** *[23, 26] Let $f$ and $g$ be two Boolean functions of order $n$ and degree $r$. Then we have $d_H(f, g) \geq 2^{n+1-r}$.*

### 2.2 Feedback Shift Registers

An $n$-stage feedback shift register (FSR) consists of $n$ binary storage cells and a feedback function $F(x_0, x_1, \ldots, x_{n-1})$ regulated by a single clock. The characteristic function of this FSR is defined as $f = F(x_0, x_1, \ldots, x_{n-1}) + x_n$, and the FSR with characteristic function $f$ is usually denoted by $\mathrm{FSR}(f)$. At every clock pulse, the current state $(s_0, s_1, \ldots, s_{n-1})$ is updated by $(s_1, s_2, \ldots, s_{n-1}, F(s_0, s_1, \ldots, s_{n-1}))$ and the bit $s_0$ is outputted. The output sequences of $\mathrm{FSR}(f)$, denoted by $G(f)$, are the $2^n$ sequences $\mathbf{s} = s_0s_1\ldots$, satisfying $s_{t+n} = F(s_t, s_{t+1}, \ldots, s_{t+n-1})$, or equivalently $f(s_t, s_{t+1}, \ldots, s_{t+n}) = 0$, for any $t \geq 0$. It is shown by Golomb that all sequences in $G(f)$ are periodic if and only if the characteristic function $f$ is

nonsingular, i.e., of the form $f = x_0 + f_0(x_1, \ldots, x_{n-1}) + x_n$. In the sequel, all characteristic functions are assumed to be nonsingular.

From an initial state $\mathbf{S}_0 = (s_0, s_1, \ldots, s_{n-1})$, after consecutive clock pulses, 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 written in the form of $C = [s_0, s_1, \ldots, s_{l-1}]_n$ or simply $C = [s_0, s_1, \ldots, s_{l-1}]$. It is easy to see that the set $\mathbb{F}_2^n$ is partitioned into disjoint cycles $C_1, C_2, \ldots, C_k$ by FSR($f$). Conversely, a partition of $\mathbb{F}_2^n$ into cycles explicitly defines an $n$-stage FSR. Hence, we can view FSR($f$) as a set of cycles.

An FSR is called a linear feedback shift register (LFSR) if its characteristic function $f$ is linear; otherwise, it is called a nonlinear feedback shift register (NFSR). With a linear characteristic function $f(x_0, x_1, \ldots, x_n) = a_0x_0 + a_1x_1 + \cdots + a_nx_n$, a univariate polynomial $c(x) = a_0 + a_1x + \cdots + a_nx^n \in \mathbb{F}_2[x]$ can be associated, that is usually called the characteristic polynomial of the LFSR. In the case of LFSRs it is more convenient to use the term characteristic polynomial rather than characteristic function.

Let $\mathbf{a}$ be a sequence and $l$ be the linear characteristic function of least order such that $\mathbf{a} \in G(l)$. Then the linear complexity of $\mathbf{a}$ is defined to be the order of $l$. The minimal polynomial of $\mathbf{a}$, denoted by $m(\mathbf{a})$, is the univariate polynomial corresponding to the linear characteristic function $l$. Let $p(x) \in \mathbb{F}_2[x]$ be a polynomial. The period of $p(x)$, denoted by $\mathrm{per}(p(x))$, is the least integer $k$ such that $p(x)|(1 + x^k)$. Some properties of the period of a polynomial can be found in [19].

For an $n$-stage FSR, the length of the periods of its output sequences is limited by $2^n$, as there are only so many different states. If this value is attained by a single sequence, it is called a de Bruijn sequence, and the FSR is called a maximum-length FSR. The unique cycle in a maximum-length FSR is called a de Bruijn cycle or a full cycle.

### 2.3 Cycle Joining Method

Let $\mathbf{S} = (s_0, s_1, \ldots, s_{n-1})$ be an FSR state. If $\mathbf{S} = (0, 0, \ldots, 0)$, the state is called zero state. The companion of $\mathbf{S}$ is defined to be $\widetilde{\mathbf{S}} = (s_0, s_1, \ldots, \overline{s}_{n-1})$, 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 there exists a companion pair $(\mathbf{S}, \widetilde{\mathbf{S}})$ such that the state $\mathbf{S}$ is on $C_1$ while its companion $\widetilde{\mathbf{S}}$ is on $C_2$. Companion pairs can be used to join cycles. If two cycles $C_1$ and $C_2$ share a companion pair $(\mathbf{S}, \widetilde{\mathbf{S}})$, then the two cycles can be joined by interchanging the predecessors of $\mathbf{S}$ and $\widetilde{\mathbf{S}}$. This is the basic idea of the cycle joining method proposed by Golomb [8]. For the application of this method, the locations of companion pairs shared by the cycles need to be found.

In [12], Jansen et al. proposed a general algorithm for joining cycles, that can be used to join the cycles of an arbitrary FSR into one full cycle. They appoint for each cycle that does not contain the zero state a representative, then they show that interchanging the predecessors of the representatives with that of their companions results in a full cycle.

Specifically, they regard a state $\mathbf{S} = (s_0, s_1, \ldots, s_{n-1})$ as a binary integer, $\mathbf{S} = \sum_{i=0}^{n-1} s_i 2^{n-1-i}$. For a cycle $C$, if it does not contain the zero state then its representative is the least state on it; otherwise, it has no representative. For a given FSR, let $C_0, C_1, \ldots, C_k$ be its cycles, where $C_0$ is the cycle containing the zero state. Let $\mathbf{S}_i$ be the cycle representative of $C_i$, and $\widetilde{\mathbf{S}}_i$ be the companion of $\mathbf{S}_i$ for $i = 1, 2, \ldots, k$. Then interchanging the predecessors of $\mathbf{S}_i$ and $\widetilde{\mathbf{S}}_i$ for $i = 1, 2, \ldots, k$, results in a full cycle.

By using this fact, Jansen et al. [12] designed an algorithm to join cycles of an arbitrary FSR. The most time-consuming portion of their algorithm is to test a state whether it is a representative or not at every step, which is usually done by traversing the cycle that contains the state. Therefore, the efficiency of the algorithm depends on the length of the longest cycle in the FSR. In order to make the algorithm efficient, they construct a class of LFSRs with only short cycles, by using the theory of LFSRs.

## 3 Cycle Joining Algorithm

In this section, we suggest another way to define cycle representatives, that has the same effect of joining cycles as in [12]. Let $C = [\mathbf{S}_0, \mathbf{S}_1, \ldots, \mathbf{S}_{l-1}]$ be a cycle , where $\mathbf{S}_i = (s_i, s_{i+1}, \ldots, s_{i+n-1})$. Write this cycle in the form of $C = [s_0, s_1, \ldots, s_{l-1}]$, and consider the longest 0-run in it. For example, the longest 0-run in the cycle $C = [1, 1, 0, 0, 0, 1, 1, 0]$ has length 3. By utilizing the longest 0-run in a cycle we give another definition of cycle representatives.

**Definition 1.** *Let $C$ be a cycle that does not contain the zero state. The cycle representative of $C$ is defined to be the state $\mathbf{S}$ on $C$ that satisfies the following two conditions,*

1. *It has the form $(*, \ldots, *, \overbrace{0, \ldots, 0}^{t}, 1)$, where $t$ is the length of the longest 0-run in $C$.*
2. *Its reverse $r\mathbf{S} = (s_{n-1}, s_{n-2}, \ldots, s_0)$ has the least value interpreted as an integer.*

The cycle that contains the zero state, has no representative, while cycles that do not contain the zero state must each have a unique representative. Clearly, in general there is more than one state satisfying Condition 1 of Definition 1, so we add Condition 2 to ensure the uniqueness of cycle representatives. Actually, Condition 2 can be replaced by any other condition such that it ensures the uniqueness. The reason that we choose explicitly this condition is that, using this condition we can efficiently test whether a given state is the representative of its cycle or not (see Algorithm 1 for details). Since this test will do many times in the final cycle joining algorithm (Algorithm 2), its efficiency strongly influence the entire efficiency.

This definition of cycle representative is almost the same as that in [12] except that it requires the representatives always ending with 1. For example, the cycle representative of $C = [(010), (101)]$ is (101), while under the original definition in [12] its representative is (010). The following Algorithm 1 is an explicit algorithm to test whether a given state is the cycle

representative or not. If it is, the algorithm returns 1; otherwise it returns 0. For simplicity, we denote the zero state by $\mathbf{0}$.

**Algorithm 1** Test for cycle representatives

```
1: INPUT: The feedback function F of the FSR, and a state S_0 to be tested.
2: OUTPUT: Whether S_0 is the cycle representative or not.
3: if lsb(S_0) = 0 then
4:     return 0;
5: end if
6: S ← The next state of S_0;
7: while S ≠ S_0 do
8:     if "S = 0" or "lsb(S) = 1 and rS < rS_0" then
9:         return 0;
10:    end if
11:    S ← The next state of S;
12: end while
13: return 1;
```

The special form of cycle representatives according to our definition can be used to speed up Algorithm 1, a property that is the direct consequence of Theorem 1.

**Theorem 1.** *Let $C$ be a cycle and $\mathbf{S} = (*, \ldots, *, \overbrace{0, \ldots, 0}^{t}, 1)$ be a state on $C$. Then none of the next $t$ states is a cycle representative.*

*Proof.* It is easy to see that the next $t$ states are not of the form of $(*, \ldots, *, \overbrace{0, \ldots, 0}^{u}, 1)$ with $u \geq t$. Hence, by the definition, none of them is a cycle representative as $u < t$. □

The following theorem shows that, with the help of cycle representatives as defined by Definition 1, the cycles of an arbitrary FSR are combined into one full cycle.

**Theorem 2.** *For a given FSR, let $C_0, C_1, \ldots, C_k$ be its cycles, where $C_0$ is the cycle containing the zero state. Let $\mathbf{S}_i$ be the cycle representative of $C_i$, and $\widetilde{\mathbf{S}}_i$ be the companion of $\mathbf{S}_i$ for $i = 1, 2, \ldots, k$. Then interchanging the predecessors of $\mathbf{S}_i$ and $\widetilde{\mathbf{S}}_i$ for $i = 1, 2, \ldots, k$, results in a full cycle.*

*Proof.* Let $t_i$ be the length of the longest 0-run in $\mathbf{S}_i$ for $i = 1, 2, \ldots, k$. Since $\mathbf{S}_i$ has the form of $\mathbf{S}_i = (*, \ldots, *, \overbrace{0, \ldots, 0}^{t_i}, 1)$, the length of the longest 0-run in $\widetilde{\mathbf{S}}_i$ is $t_i + 1$. By the definition of cycle representative, any state on $C_i$ can not have more than $t_i$ consecutive 0s. Therefore, $\widetilde{\mathbf{S}}_i$ is on some cycle other than $C_i$. Assume $\widetilde{\mathbf{S}}_i$ is on the cycle $C_j$ with $j \neq i$. Since $\widetilde{\mathbf{S}}_i$ is of the form $\widetilde{\mathbf{S}}_i = (*, \ldots, *, \overbrace{0, \ldots, 0}^{t_i+1})$, it can not be the cycle representative of $C_j$, that is, $\widetilde{\mathbf{S}}_i \neq \mathbf{S}_j$. Again,

by the definition of cycle representative, we get $t_j > t_i$. Let $G$ be the directed graph that takes $C_0, C_1, \ldots, C_k$ as its nodes, and there is a directed edge from $C_i$ to $C_j$ if and only if $\widetilde{\mathbf{S}}_i$ is on $C_j$. Then by the above discussion, $G$ is a directed tree with root $C_0$. This tree represents a choice of companion pairs that repeatedly join two cycles into one, ending with exactly one cycle, i.e., a full cycle. □

The following Algorithm 2 is a cycle joining algorithm with the concrete steps for generating full cycles from an arbitrary FSR.

**Algorithm 2** Generation of the full cycle

1: INPUT: The feedback function $F(x_0, x_1, \ldots, x_{n-1})$ of the FSR, and the initial state $\mathbf{S}_0 = (s_0, s_1, \ldots, s_{n-1})$.
2: OUTPUT: The full cycle $C = [\mathbf{S}_0, \mathbf{S}_1, \ldots, \mathbf{S}_{2^n-1}]$.
3: **for** $i = 0, 1, \ldots, 2^n - n - 1$ **do**
4: **if** $(s_{i+1}, \ldots, s_{i+n-1}, 1)$ is a cycle representative **then**
5: $\mathbf{S}_{i+1} = (s_{i+1}, \ldots, s_{i+n-1}, F(s_{i+1}, s_{i+2}, \ldots, s_{i+n-1}) + 1)$
6: **else**
7: $\mathbf{S}_{i+1} = (s_{i+1}, \ldots, s_{i+n-1}, F(s_{i+1}, s_{i+2}, \ldots, s_{i+n-1}))$
8: **end if**
9: **end for**
10: return $C$

This algorithm complements the value of the feedback function if and only if there is a cycle representative amongst the two possible successors $(s_{i+1}, \ldots, s_{i+n-1}, 0)$ and $(s_{i+1}, \ldots, s_{i+n-1}, 1)$. As the representatives defined in this paper always end with 1, the state $(s_{i+1}, \ldots, s_{i+n-1}, 0)$ can never be a representative. Therefore, we just need to test whether $(s_{i+1}, \ldots, s_{i+n-1}, 1)$ is a cycle representative. If the cycle representatives defined in [12] would be used, then the two possible successors would both have to be tested. So the efficiency of the new cycle joining algorithm is twice of that proposed in [12]. Since the testing is done by traversing the cycle that contains the state needing to be tested (see Algorithm 1), it requires $2n$ bits of storage and at most $l$ FSR shifts for the generation of the next state on the full cycle, where $l$ is the length of the longest cycle of the FSR.

If the original definition of cycle representative in [12] is used, then both possible successors need to be tested. However, if the first state under test is a cycle representative then the other state need not be tested. In this case the gain of the new algorithm appears to be less than a factor of two. It is easily shown that this case occurs with very low probability and in most cases both states are not representatives. It is known that an $n$-stage FSR generates no more than $Z(n) = \frac{1}{n}\sum_{d|n} \phi(d) 2^{\frac{n}{d}}$ cycles [25]. So the number of bridging states (a state for which one of its successors is a cycle representative) is no more than $2Z(n) - 2$. This means that for the other $2^n - (2Z(n) - 2)$ states, the possible successors are not cycle representatives. For those non-bridging states, if the original definition is used, both successors need to be tested. The fraction of the non-bridging states is at least $\frac{2^n - (2Z(n)-2)}{2^n} = O(1 - \frac{2}{n})$.

## 4 NFSRs with Short Cycles

The performance of the two cycle joining algorithms proposed in [12] and in this paper depend heavily on the length of the longest cycle in the FSR. Evidently, FSRs with only short cycles are needed. In [12], a class of short-cycle LFSRs was constructed by using the theory of LFSRs. In this section, we use the composition operator $*$ to construct a large class of NFSRs.

Let $f(x_0, x_1, \ldots, x_n)$ be a characteristic function of order $n$ and $\mathbf{a} = a_0a_1\ldots$, be a sequence. Define $\theta(f)$ to be the mapping on sequences: $\theta(f)(\mathbf{a}) = \mathbf{b}$, where $\mathbf{b}$ is determined by $b_i = f(a_i, a_{i+1}, \ldots, a_{i+n})$. Let $\theta(f)^{-1}(\mathbf{b})$ be the set of sequences whose image is $\mathbf{b}$ under $\theta(f)$, that is, $\theta(f)^{-1}(\mathbf{b}) = \{\mathbf{a} \mid \theta(f)(\mathbf{a}) = \mathbf{b}\}$. Then it can be verified that $\theta(f)^{-1}(\mathbf{b})$ contains $2^n$ sequences and we have $\theta(f)^{-1}(\mathbf{0}) = G(f)$. When the operation $*$ is considered, we have $\theta(f*g) = \theta(f)\theta(g)$, which implies that,

$$G(f * g) = \theta(f * g)^{-1}(\mathbf{0}) = \theta(g)^{-1}\theta(f)^{-1}(\mathbf{0}) = \theta(g)^{-1}(G(f)).$$

The above expression shows that the sequences in $G(f * g)$ are obtained by applying $\theta(g)^{-1}$ to $G(f)$. This gives a method to investigate the cycle structure of FSR$(f * g)$. In the following, we consider the case that $f$ is a nonlinear function and $g$ is linear. This is to find $f$ and $g$ such that $G(f * g)$ contains only sequences of small period, which means that FSR$(f * g)$ generates only short cycles. Some properties of $\theta(f)^{-1}$, especially when $f$ is linear, were given in [24]. We recall some of them without proofs in the following lemma.

**Lemma 2.** *[24] Let $l$ be a linear characteristic function and $c(x)$ be the univariate polynomial that corresponding to $l$. Let $\mathbf{a}$ be a periodic sequence and $m(\mathbf{a})$ be the minimal polynomial of $\mathbf{a}$.*

1. *If* $\gcd(c(x), m(\mathbf{a})) = 1$, *then* $\theta(l)^{-1}(\mathbf{a}) = \mathbf{b} + G(l)$ *for some* $\mathbf{b} \in G(m(\mathbf{a}))$ *with* $m(\mathbf{b}) = m(\mathbf{a})$.
2. *If $c(x)$ is irreducible and $m(\mathbf{a}) = h(x)c(x)^e$, $e \geq 1$ with* $\gcd(h(x), c(x)) = 1$, *then* $m(\mathbf{b}) = h(x)c(x)^{e+1}$ *for all* $\mathbf{b} \in \theta(l)^{-1}(\mathbf{a})$.
3. *The least common multiple* $\mathrm{lcm}\{m(\mathbf{b}) \mid \mathbf{b} \in \theta(l)^{-1}(\mathbf{a})\} = m(\mathbf{a})c(x)$.

The next lemma provides a property on the periods of polynomials that will be used later.

**Lemma 3.** *If the integer $t$ satisfies $1 \leq t \leq 2^m$, then* $\mathrm{per}((1 + x^t)(1 + x^{2^m})) = \mathrm{lcm}(t, 2^{m+1})$.

*Proof.* Let $t = 2^u v$, where $v$ is an odd number. Then $(1 + x^t)(1 + x^{2^m}) = (1 + x^v)^{2^u}(1 + x)^{2^m} = (1 + x)^{2^u}(1 + x + \cdots + x^{v-1})^{2^u}(1 + x)^{2^m} = (1 + x + \cdots + x^{v-1})^{2^u}(1 + x)^{2^m+2^u}$. Since $\gcd((1 + x + \cdots + x^{v-1})^{2^u}, (1 + x)^{2^m+2^u}) = 1$, we have $\mathrm{per}((1 + x + \cdots + x^{v-1})^{2^u}(1 + x)^{2^m+2^u}) = \mathrm{lcm}(\mathrm{per}((1 + x + \cdots + x^{v-1})^{2^u}), \mathrm{per}((1 + x)^{2^m+2^u})$. Clearly $\mathrm{per}(1 + x + \cdots + x^{v-1}) = v$. Hence, $\mathrm{per}((1 + x + \cdots + x^{v-1})^{2^u}) = v2^u = t$. Since $1 \leq 2^u \leq 2^m$, we have $\mathrm{per}((1 + x)^{2^m+2^u}) = 2^{m+1}$. So that $\mathrm{per}((1 + x^t)(1 + x^{2^m})) = \mathrm{lcm}(t, 2^{m+1})$. □

By using this lemma, a class of short-cycle FSRs is constructed.

**Theorem 3.** *Let $f$ be a characteristic function of order $m$. Then for any sequence $\mathbf{s} \in G(f * (x_0 + x_{2^m}))$ we have $\mathrm{per}(\mathbf{s}) \leq 2^{2m+1}$.*

*Proof.* Let $\mathbf{s}$ be a sequence in $G(f*(x_0+x_{2^m}))$. Then we have $\theta(f*(x_0+x_{2^m}))(\mathbf{s}) = \theta(f)\theta(x_0+x_{2^m})(\mathbf{s}) = \mathbf{0}$, which implies $\theta(x_0+x_{2^m})(\mathbf{s}) \in \theta(f)^{-1}(\mathbf{0})$. Since $\theta(f)^{-1}(\mathbf{0}) = G(f)$, there exists a sequence $\mathbf{a} \in G(f)$ such that $\theta(x_0+x_{2^m})(\mathbf{s}) = \mathbf{a}$. Therefore, we have $\mathbf{s} \in \theta(x_0+x_{2^m})^{-1}(\mathbf{a})$. According to Case 3 of Lemma 2, we have $m(\mathbf{s})|m(\mathbf{a})(1+x^{2^m})$. Let $0 \leq t \leq 2^m$ be the period of $\mathbf{a}$, then $m(\mathbf{a})|(1+x^t)$. Hence, $m(\mathbf{s})|(1+x^t)(1+x^{2^m})$. According to Lemma 3, $\mathrm{per}((1+x^t)(1+x^{2^m})) = \mathrm{lcm}(t, 2^{m+1})$. Therefore, we get that $\mathrm{per}(m(\mathbf{s})) \leq \mathrm{lcm}(t, 2^{m+1}) \leq t2^{m+1} \leq 2^{2m+1}$. □

The order of $f*(x_0+x_{2^m})$ is $n = m + 2^m$. By Theorem 3, the lengths of the cycles in $\mathrm{FSR}(f*(x_0+x_{2^m}))$ are less than $2n^2$, which means that $\mathrm{FSR}(f*(x_0+x_{2^m}))$ generates only short cycles. Next, we investigate the case that $f$ is the characteristic function of an $m$-stage maximum length FSR. In this case, much more useful results are obtained.

**Theorem 4.** *Let $f$ be the characteristic function of an $m$-stage maximum length FSR, and $l$ be the linear Boolean function corresponding to $(1+x)^r$. Then all the sequences in $G(f*l)$ have the same period, and the period satisfies $2^{\lceil \log(2^{m-1}+m+r) \rceil} \leq T \leq 2^{\lceil \log(2^m+r-1) \rceil}$.*

*Proof.* Since $\mathrm{FSR}(f)$ is a maximum length FSR, the sequences in $G(f)$ are all de Bruijn sequences. These de Bruijn sequences have the same linear complexity and the same minimal polynomial. Let $\Lambda$ be their linear complexity. Then their minimal polynomial is given by $(1+x)^\Lambda$. It is well known that $\Lambda$ satisfies $2^{m-1}+m \leq \Lambda \leq 2^m - 1$ (see [1]).

According to Lemma 2, the sequences in $\theta(l)^{-1}(G(f))$ have the same minimal polynomial $(1+x)^{\Lambda+r}$, which implies that they have the same period. Because $G(f*l) = \theta(l)^{-1}(G(f))$, the sequences in $G(f*l)$ have the same period. Since $\Lambda$ satisfies $2^{m-1}+m \leq \Lambda \leq 2^m - 1$, we have

$$2^{\lceil \log(2^{m-1}+m+r) \rceil} \leq \mathrm{per}((1+x)^{\Lambda+r}) \leq 2^{\lceil \log(2^m+r-1) \rceil}.$$

Therefore, the period of the sequences in $G(f*l)$ satisfies $2^{\lceil \log(2^{m-1}+m+r) \rceil} \leq T \leq 2^{\lceil \log(2^m+r-1) \rceil}$. □

By properly choosing the values of the parameters, a class of short-cycle FSRs is obtained.

**Corollary 1.** *Let $\mathrm{FSR}(f*l)$ be an $n$-stage FSR, where $f$ is the characteristic function of an $m$-stage maximum length FSR and $l$ is the linear Boolean function corresponding to $(1+x)^{n-m}$. Then:*

1. *If $m = \lceil \log n \rceil$, then for any sequence $\mathbf{s} \in G(f*l)$ we have $\mathrm{per}(\mathbf{s}) = 2^{m+1}$.*
2. *If $m = \lfloor \log n \rfloor$, then for any sequence $\mathbf{s} \in G(f*l)$ we have $\mathrm{per}(\mathbf{s}) = 2^{m+1}$ or $2^{m+2}$.*

*In either case, the length of the cycles in* $\mathrm{FSR}(f * l)$ *is at most* $4n$.

*Proof.* Define $r = n - m$. If $m = \lceil \log n \rceil$, then we have $2^{m-1} + 1 \leq n \leq 2^m$ and $2^{m-1} - m + 1 \leq r \leq 2^m - m$. By simple calculation we get that,

$$2^m + 1 \leq 2^{m-1} + m + r \leq 2^m + 2^{m-1},$$
$$2^m + 2^{m-1} - m \leq 2^m + r - 1 \leq 2^{m+1} - m - 1.$$

By Theorem 4, the period of the sequences in $G(f * l)$ satisfies,

$$2^{m+1} = 2^{\lceil \log(2^m+1) \rceil} \leq T \leq 2^{\lceil \log(2^{m+1}-m-1) \rceil} = 2^{m+1}.$$

Since $2^{m-1} + 1 \leq n \leq 2^m$ we know $2n \leq 2^{m+1} \leq 4(n-1)$, which implies that the cycles in $\mathrm{FSR}(f * l)$ have length $\leq 4n$.

If $m = \lfloor \log n \rfloor$ , Then we have $2^m \leq n \leq 2^{m+1} - 1$ and $2^m - m \leq r \leq 2^{m+1} - m - 1$. By simple calculation we get that,

$$2^m + 2^{m-1} \leq 2^{m-1} + m + r \leq 2^{m+1} + 2^{m-1} - 1,$$
$$2^{m+1} - m - 1 \leq 2^m + r - 1 \leq 2^{m+1} + 2^m - m - 2.$$

By Theorem 4, the period of the sequences in $G(f * l)$ satisfies,

$$2^{m+1} = 2^{\lceil \log(2^m+2^{m-1}) \rceil} \leq T \leq 2^{\lceil \log(2^{m+1}+2^m-m-2) \rceil} = 2^{m+2}.$$

Note that the periods of the sequences in $G(f * l)$ must be a power of 2, so we have $T = 2^{m+1}$ or $T = 2^{m+2}$. By $2^m \leq n \leq 2^{m+1} - 1$ we know $n + 1 \leq 2^{m+1} \leq 2n$, which implies that the cycles in $\mathrm{FSR}(f * l)$ have length $\leq 4n$. □

The $n$-stage FSRs constructed in Corollary 1 generates only short cycles of lengths at most $4n$. An example of such FSRs is given in Example 1. These FSRs are good candidates for the cycle joining algorithms proposed in [12] and in this paper. In the following, we consider the number of full cycles constructed from them by using the cycle joining algorithms. We use the fact about minimum Hamming distance of Reed-Muller codewords, which is suggested by Jansen et al. [12]. We proceed by proving that two different characteristic functions give rise to two different de Bruijn sequences. First, we need the following lemma for the proof of the theorem.

**Lemma 4.** *Let* $f$ *be a Boolean function and* $l$ *be a linear Boolean function. Then* $\deg f * l = \deg f$.

*Proof.* Let $l = x_{k_1} + x_{k_2} + \cdots + x_{k_u}$ be the ANF of $l$, where $k_1, k_2, \cdots, k_u$ are nonnegative integers with $k_1 < k_2 < \cdots < k_u$. Let $t = x_{i_1} x_{i_2} \cdots x_{i_v}$ be a term of $f$, then $t * l = (x_{k_1+i_1} + x_{k_2+i_1} + \cdots + x_{k_u+i_1})(x_{k_1+i_2} + x_{k_2+i_2} + \cdots + x_{k_u+i_2}) \cdots (x_{k_1+i_v} + x_{k_2+i_v} + \cdots + x_{k_u+i_v})$, implying $\deg t * l \leq \deg t$. Hence, $\deg f * l \leq \deg f$. We associate each term of $f$ with an integer: $N(x_{i_1} x_{i_2} \cdots x_{i_v}) =$

$2^{i_1}+2^{i_2}+\cdots+2^{i_v}$. Let $x_{j_1}x_{j_2}\cdots x_{j_d}$ be the term of $f$ such that $N(x_{j_1}x_{j_2}\cdots x_{j_d})$ is the smallest among all the terms of $f$ with degree $d$, where $d$ is the degree of $f$. The lemma is proved by showing that $x_{k_1+j_1}x_{k_1+j_2}\cdots x_{k_1+j_d}$ is a term of $f*l$. First note that $(x_{j_1}x_{j_2}\cdots x_{j_d})*l$ contains the term $x_{k_1+j_1}x_{k_1+j_2}\cdots x_{k_1+j_d}$. Let $x_{i_1}x_{i_2}\cdots x_{i_k}$ be a term of $f$ such that $x_{i_1}x_{i_2}\cdots x_{i_k}\neq x_{j_1}x_{j_2}\cdots x_{j_d}$. It then has to be shown that $\tau$, with $\tau=(x_{i_1}x_{i_2}\cdots x_{i_k})*l$ does not contain the term $x_{k_1+j_1}x_{k_1+j_2}\cdots x_{k_1+j_d}$. If $k\neq d$, then $\tau$ only contains terms of degree $k$, and consequently does not contain the term $x_{j_1}x_{j_2}\cdots x_{j_d}$. If $k=d$, then by the definition of $N(x_{j_1}x_{j_2}\cdots x_{j_d})$, we have $2^{i_1}+2^{i_2}+\cdots+2^{i_k}>2^{j_1}+2^{j_2}+\cdots+2^{j_d}$. Note that $\tau$ only contains terms whose associated integers are at least $2^{i_1}+2^{i_2}+\cdots+2^{i_k}$, and as a consequence, $\tau$ does not contain the term $x_{k_1+j_1}x_{k_1+j_2}\cdots x_{k_1+j_d}$. □

Let $f$ be a Boolean function of order $m$. Some necessary conditions for FSR($f$) being a maximum-length FSR are given in [8]. One of these conditions is that $\deg f=m-1$. In the following it is assumed that $f$ satisfies this condition. We are now able to prove the uniqueness of the generated de Bruijn sequences.

**Theorem 5.** *Let $n$ be a positive integer. Define $m=\lceil\log n\rceil$ and $r=n-m$. Let $f_1$ and $f_2$ be the characteristic functions of two $m$-stage maximum-length FSRs with $f_1\neq f_2$, and $l$ be the linear Boolean function corresponding to $(1+x)^r$. If a cycle joining algorithm (proposed in [12] or in this paper) is applied to* FSR($f_1*l$) *and* FSR($f_2*l$)*, then the two resulting de Bruijn sequences are different.*

*Proof.* By Lemma 4, we have $\deg f_1*l=\deg f_1=m-1$ and $\deg f_2*l=\deg f_2=m-1$. Since $\mathrm{ord}(f_1*l)=\mathrm{ord}(f_2*l)=n$, Lemma 1 implies $d_H(f_1*l,f_2*l)\geq 2^{n-m+2}$. Let FSR($h_1$) and FSR($h_2$) be the two maximum-length FSRs derived from FSR($f_1*l$) and FSR($f_2*l$) by using the cycle joining algorithm. According to Theorem 4, FSR($f_1*l$) only generates cycles of length $2^{m+1}$. Hence, there are $\frac{2^n}{2^{m+1}}=2^{n-m-1}$ cycles in this FSR. Every time two cycles in this FSR are joined together in the process of the cycle joining, the weight of the corresponding characteristic function is changed by 4. Hence, $d_H(f_1*l,h_1)\leq 4\left(2^{n-m-1}-1\right)=2^{n-m+1}-4$. Similarly, $d_H(f_2*l,h_2)\leq 2^{n-m+1}-4$. It now follows that

$$\begin{aligned} d_H(h_1,h_2) &\geq d_H(f_1*l,f_2*l)-d_H(f_1*l,h_1)-d_H(f_2*l,h_2)\\ &\geq 2^{n-m+2}-\left(2^{n-m+1}-4\right)-\left(2^{n-m+1}-4\right)\\ &=8. \end{aligned}$$

□

The reader can verify that Theorem 5 also holds for $m=\lfloor\log n\rfloor$. The number of $m$-stage maximum length FSRs is $2^{2^{m-1}-m}$ [3]. Theorem 5 tells us that $2^{2^{m-1}-m}$ de Bruijn sequences can be constructed based on the short-cycle FSRs of Corollary 1. As $m=\lceil\log n\rceil$ (or $m=\lfloor\log n\rfloor$), we have $2^{2^{m-1}-m}=O(2^{\frac{n}{2}-\log n})$, which implies that the size of the class of de Bruijn sequences for this construction grows exponentially with the order $n$. Table 1 lists the results for the two cases (i) $m=\lceil\log n\rceil$ and (ii) $m=\lfloor\log n\rfloor$.

Table 1: Different settings and the corresponding results

| | the scope of $n$ | cycle length len | the ratio len/$n$ | # de Bruijn sequences |
|---|---|---|---|---|
| $m = \lceil \log n \rceil$ | $2^{m-1} + 1 \leq n \leq 2^m$ | $\text{len} = 2^{m+1}$ | $\frac{\text{len}}{n} \leq 4$ | $2^{2^{m-1}-m}$ |
| $m = \lfloor \log n \rfloor$ | $2^m \leq n \leq 2^{m+1} - 1$ | $\text{len} = 2^{m+1}$ or $2^{m+2}$ | $\frac{\text{len}}{n} \leq 4$ | $2^{2^{m-1}-m}$ |

The integer $m$ is purposely chosen close to the value $\log n$, because, on the one hand, a smaller $m$ results in fewer choices of $f$, which implies a smaller size of the de Bruijn sequences; on the other hand, a bigger $m$ results in longer cycles in $\text{FSR}(f * l)$, which reduces the efficiency of the cycle joining algorithm.

At the end of this section, we present an example to illustrate that the two definitions of cycle representatives in [12] and in this paper are essentially different. This example is also a verification of Corollary 1.

**Example 1.** *Let $f = x_0 + 1 + x_2$ be the characteristic function of the unique 2-stage maximum-length FSR, and $g = f * (x_0 + x_4) = x_0 + x_2 + x_4 + 1 + x_6$. By Corollary 1, there are 8 cycles in $\text{FSR}(g)$, all of them are of length 8.*

$$C_0 = [000000, 000001, 000011, 000110, 001100, 011000, 110000, 100000],$$
$$C_1 = [000010, 000100, 001001, 010010, 100100, 001000, 010000, \underline{100001}],$$
$$C_2 = [000101, 001011, 010111, 101110, 011100, 111000, \underline{110001}, 100010],$$
$$C_3 = [000111, 001110, 011101, 111010, 110100, 101000, \underline{010001}, 100011],$$
$$C_4 = [001010, 010101, 101011, 010110, 101100, \underline{011001}, 110010, 100101],$$
$$C_5 = [001101, 011010, 110101, 101010, 010100, \underline{101001}, 010011, 100110],$$
$$C_6 = [001111, 011111, 111111, 111110, 111100, \underline{111001}, 110011, 100111],$$
$$C_7 = [011011, 110111, 101111, 011110, 111101, 111011, 110110, \underline{101101}].$$

*The first state in each cycle is the cycle representative defined in [12], and the underlined state in each cycle is the cycle representative defined in this paper. Let $G$ be the directed graph that take $C_0, C_1, \ldots, C_7$ as its nodes, and there is a directed edge from $C_i$ to $C_j$ if and only if the companion of the representative of $C_i$ is located on $C_j$. Then $G$ is a directed tree with root $C_0$ The two trees are shown in Figure 1, where the left one is based on the definition of cycle representative in [12] and the right one is based on the definition in this paper.*

*Note that the height of the directed tree based on the definition of cycle representative in this paper does not exceed $n+1$. However, it is currently unclear to the authors whether this property holds under the definition in [12].*

## 5 Symmetric Shift Registers

Another class of NFSRs with short cycles only comes from symmetric FSRs. A symmetric Boolean function is a Boolean function whose value does not depend on the permutation of

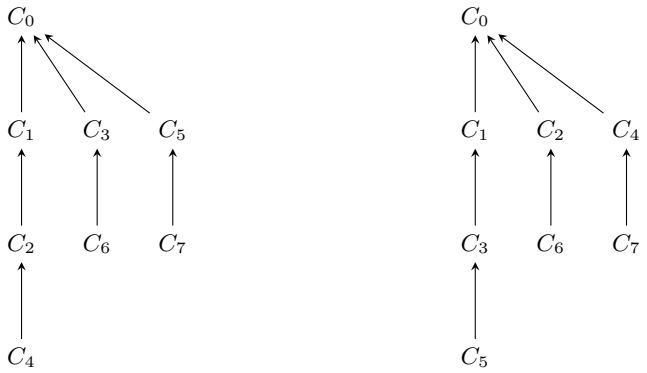


Figure 1: The adjacency trees

its input bits, that is, it depends only on the Hamming weight $w_H$ of the input. It is easily verified that, the symmetric Boolean functions in the variables $x_1, x_2, \ldots, x_{n-1}$ form a vector space of dimension $n$, with $E_k(x_1, x_2, \ldots, x_{n-1}), k = 0, 1, \ldots, n-1$ being the basis, where $E_k$ is defined to be the symmetric Boolean function such that: $E_k(s_1, s_2, \ldots, s_{n-1}) = 1$ if and only if $w_H(s_1, s_2, \ldots, s_{n-1}) = k$, and these $E_k$ are called elementary symmetric Boolean functions. Let $h(x_1, x_2, \ldots, x_{n-1})$ be a symmetric Boolean function, from the above discussion we know that there exists a unique subset $M \subset \{0, 1, \ldots, n-1\}$ such that $h(x_1, x_2, \ldots, x_{n-1}) = \sum_{k \in M} E_k$. We call the subset $M$ that determined by $h$ the index set of $h$, and denote it by $\mathrm{Ind}(h)$.

An FSR($f$) is called a symmetric FSR if $f$ has the form $f = x_0 + h(x_1, x_2, \ldots, x_{n-1}) + x_n$, where $h$ is a symmetric Boolean function.

### 5.1 Scattered Symmetric FSRs

A symmetric Boolean function is called scattered symmetric if for any $i \in \mathrm{Ind}(h)$ it holds that $i - 1 \notin \mathrm{Ind}(h)$, that is, the index set $\mathrm{Ind}(h)$ does not include any two consecutive integers. For example, there are three scattered symmetric Boolean functions of one variable, corresponding to $\mathrm{Ind}(h) = \emptyset, \{0\}, \{1\}$ respectively. Although symmetric, $\mathrm{Ind}(h) = \{0, 1\}$ does not correspond to a scattered symmetric function. Similarly, there are five scattered symmetric Boolean functions of two variables, corresponding to $\mathrm{Ind}(h) = \emptyset, \{0\}, \{1\}, \{2\}, \{0, 2\}$ respectively. The remaining three symmetric functions $\mathrm{Ind}(h) = \{0, 1\}, \{1, 2\}, \{0, 1, 2\}$ do not correspond to a scattered symmetric function. An FSR($x_0 + h + x_n$) is called a scattered symmetric FSR if $h(x_1, \ldots, x_{n-1})$ is scattered symmetric.

It was shown in [29] that, an $n$-stage scattered symmetric FSR only generates cycles of length at most $n+1$. Therefore, the cycle joining algorithms are very fast if applied to scattered symmetric FSRs. The following theorem gives the size of scattered symmetric FSRs. Counting the number of scattered symmetric Boolean functions of $n$ variables is equivalent to counting the number of binary strings of length $n+1$ with no two ones adjacent. This problem was solved

by Muir [22]. Here, the counting exercise is solved in the next theorem by obtaining a recursive relation in the number of scattered symmetric Boolean functions, following directly from the index set definition.

**Theorem 6.** *The number of $n$-stage scattered symmetric FSRs is*

$$\frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{n+2} - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^{n+2}.$$

*Proof.* Let $A(n), n > 1$ be the set of symmetric Boolean functions in the variables $x_1, x_2, \ldots, x_{n-1}$ such that: for any $i \in \mathrm{Ind}(h)$ we have $i-1 \notin \mathrm{Ind}(h)$. The symmetric Boolean functions in $A(n)$ can be divided into two classes, those satisfying $0 \notin \mathrm{Ind}(h)$ and those satisfying $0 \in \mathrm{Ind}(h)$. The first class contains $|A(n-1)|$ elements, as the remaining number of weight functions is decreased by one. The second class contains $|A(n-2)|$ elements, because the remaining number of weight functions is decreased by two, namely both indices 0 and 1. Therefore, we have $|A(n)| = |A(n-1)| + |A(n-2)|$. This implies $\{|A(n)|\}$ is a Fibonacci sequence with initial values $|A(2)| = 3$ and $|A(3)| = 5$. By Binet's Fibonacci number formula, we get that

$$|A(n)| = \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^{n+2} - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^{n+2}.$$

This completes the proof. □

If the cycle joining algorithms (proposed in [12] and in this paper) are applied to different symmetric FSRs, the resulting de Bruijn sequences may be the same. The following two theorems show the conditions under which this happens. In the proofs of the theorems, the symmetric difference of two sets $A$ and $B$ is used, that is defined as $A \triangle B = \{x \mid x \in A \cup B, x \notin A \cap B\}$.

**Theorem 7.** *Let* $\mathrm{FSR}(x_0 + h_1 + x_n)$ *and* $\mathrm{FSR}(x_0 + h_2 + x_n)$ *be two symmetric FSRs, and $C_1$ and $C_2$ be the two resulting de Bruijn cycles by applying the cycle joining algorithm of [12] to the two FSRs respectively. Then $C_1 = C_2$ if and only if* $\mathrm{Ind}(h_1) \triangle \mathrm{Ind}(h_2) \subseteq \{0, n-1\}$.

*Proof.* Suppose $\mathrm{Ind}(h_1) \triangle \mathrm{Ind}(h_2) \subseteq \{0, n-1\}$. We consider only the case of $\mathrm{Ind}(h_1) \triangle \mathrm{Ind}(h_2) = \{0\}$, the other cases can be handled similarly. Without loss of generality, assume $0 \in \mathrm{Ind}(h_1)$ and $0 \notin \mathrm{Ind}(h_2)$. Let $D_1$ be the cycle in $\mathrm{FSR}(x_0+h_1+x_n)$ that contains the zero state, $D_2$ be the cycle in $\mathrm{FSR}(x_0+h_2+x_n)$ that contains the zero state, and $D_3$ be the cycle in $\mathrm{FSR}(x_0+h_2+x_n)$ that contains the state $(0, \ldots, 0, 1)$. It can be seen that the two FSRs generate the same cycles except for $D_1$, $D_2$ and $D_3$, i.e.

$$\mathrm{FSR}(x_0 + h_1 + x_n) \setminus \{D_1\} = \mathrm{FSR}(x_0 + h_2 + x_n) \setminus \{D_2, D_3\}.$$

By applying the cycle joining algorithm, the two cycles $D_2$ and $D_3$ are joined together and result in $D_1$, implying that the two full cycles $C_1$ and $C_2$ are the same.

Suppose $\mathrm{Ind}(h_1) \triangle \mathrm{Ind}(h_2) \not\subseteq \{0, n-1\}$. Without loss of generality, assume there exists some integer $0 < i < n-1$ such that $i \in \mathrm{Ind}(h_1)$ and $i \notin \mathrm{Ind}(h_2)$. Consider the state $\mathbf{V} = (1, \overbrace{1, \ldots, 1}^{i}, 0, \ldots, 0)$. Clearly, this state has different successors in $\mathrm{FSR}(x_0 + h_1 + x_n)$ and $\mathrm{FSR}(x_0 + h_2 + x_n)$. Moreover, it can be verified (Theorem 3 of [12]) that $\mathbf{V}$ cannot be the representative of its cycles (under the definition of cycle representative in [12]). This implies that the two successors of $\mathbf{V}$ in $C_1$ and $C_2$ are different. As a consequence, $C_1$ and $C_2$ are two different full cycles. □

**Theorem 8.** *Let* $\mathrm{FSR}(x_0 + h_1 + x_n)$ *and* $\mathrm{FSR}(x_0 + h_1 + x_n)$ *be two symmetric FSRs, and* $C_1$ *and* $C_2$ *be the two resulting de Bruijn cycles by applying Algorithm 2 to them respectively. Then* $C_1 = C_2$ *if and only if* $\mathrm{Ind}(h_1) \triangle \mathrm{Ind}(h_2) \subseteq \{0, n-1\}$.

*Proof.* The proof of the sufficiency part is the same as that of Theorem 7. In the following, we show the necessity part. Since the two weights 0 and $n-1$ have no effect on the resulting full cycles, we assume that both $\mathrm{Ind}(h_1)$ and $\mathrm{Ind}(h_2)$ do not contain 0 and $n-1$.

Suppose $\mathrm{Ind}(h_1) \triangle \mathrm{Ind}(h_2) \not\subseteq \{0, n-1\}$. Without loss of generality, it is assumed that there exists an integer $0 < i < n-1$ such that $i \in \mathrm{Ind}(h_1)$ and $i \notin \mathrm{Ind}(h_2)$. Consider the following two cases, $i = 1$ and $i > 1$. If $i = 1$, then clearly, the state $(0, \ldots, 0, 1)$ has different successors in $\mathrm{FSR}(x_0 + h_1 + x_n)$ and $\mathrm{FSR}(x_0 + h_2 + x_n)$, and it must be the representative of its cycle (under the definition of cycle representative in this paper). This implies that $(0, \ldots, 0, 1)$ has different successors in $C_1$ and $C_2$. Therefore, $C_1$ and $C_2$ are two different full cycles. If $i > 1$, then the state $(0, \ldots, 0, \overbrace{1, \ldots, 1}^{i})$ has different successors in $\mathrm{FSR}(x_0 + h_1 + x_n)$ and $\mathrm{FSR}(x_0 + h_2 + x_n)$, and it can not be the representative of its cycle. This implies that this state has different successors in $C_1$ and $C_2$. Therefore, $C_1$ and $C_2$ are two different full cycles. □

It should be noted that Theorem 7 and Theorem 8 hold in general for *symmetric* FSRs. Theorem 8 directly implies the size of the class of de Bruijn sequences, constructed by applying the cycle joining algorithm to *scattered symmetric* FSRs. Consider the set of scattered symmetric FSRs, $\{\mathrm{FSR}(x_0 + h + x_n) \mid 0 \notin \mathrm{Ind}(h), n-1 \notin \mathrm{Ind}(h)\}$. Theorem 6 implies that there are $\frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n$ FSRs in this set. According to Theorem 8, if Algorithm 2 is applied, the resulting de Bruijn sequences are different from each other. So $\frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n - \frac{1}{\sqrt{5}}\left(\frac{1-\sqrt{5}}{2}\right)^n = O\left(2^{(\log(1+\sqrt{5})-1)n}\right)$ unique de Bruijn sequences can be constructed from scattered symmetric FSRs. As $\log(1+\sqrt{5}) - 1$ is approximately 0.6942, the size of the class of de Bruijn sequences constructed is very large. Note that scattered symmetric FSRs generate cycles of length at most $n+1$. Hence, generating the next state in the full cycle by Algorithm 2 requires no more than $n+1$ FSR shifts.

## 5.2 The cycle structure of symmetric FSRs

In the rest of this section, we study in detail the cycle structure and the adjacency relations of the cycles in symmetric FSRs (especially scattered symmetric FSRs), with the aim of further improving the cycle joining algorithm.

Up to now, many results about the cycle structure of symmetric FSRs have been obtained [13,29,30]. By the existing results, the symmetric FSRs have layered cycle structure, that is, the cycles of a symmetric FSR can be classified into different layers according to the weights of the states on them. The proof of the following Lemma can be found in [13,30], but for completeness we includes a concise proof in our notations here.

**Lemma 5.** *[13, 30] Let* $\mathrm{FSR}(x_0 + h + x_n)$ *be a symmetric FSR. If there exists some integer* $0 \le i \le n-1$ *such that* $i \notin \mathrm{Ind}(h)$*, then for any cycle of* $\mathrm{FSR}(x_0+h+x_n)$ *either it contains only states of weight* $\le i$ *or it contains only states of weight* $> i$*. In other words, it can not contain both states of weight* $\le i$ *and states of weight* $> i$*.*

*Proof.* Let $C$ be a cycle in $\mathrm{FSR}(x_0 + h + x_n)$. Let $(s_0, s_1, \ldots, s_{n-1})$ be a state on $C$, and $(s_1, s_2, \ldots, s_n)$ be its next state on $C$. By the recursive relation we have,

$$s_n = s_0 + h(s_1, \ldots, s_{n-1}).$$

Our discussions are divided into two cases:

The case of $w_H(s_0, s_1, \ldots, s_{n-1}) \le i$: Suppose $w_H(s_0, s_1, \ldots, s_{n-1}) < i$ then it must be that $w_H(s_1, s_2, \ldots, s_n) \le i$. In the following we assume $w_H(s_0, s_1, \ldots, s_{n-1}) = i$. If now we have $s_0 = 0$, then $w_H(s_1, \ldots, s_{n-1}) = i$. Because $i \notin \mathrm{Ind}(h)$, we have $h(s_1, \ldots, s_{n-1}) = 0$. Hence $s_n = s_0 + h(s_1, \ldots, s_{n-1}) = 0$, which implies that $w_H(s_1, s_2, \ldots, s_n) = i$. If $s_0 = 1$, then $w_H(s_1, s_2, \ldots, s_n) \le w_H(s_0, s_1, \ldots, s_{n-1}) = i$. No matter which is the case, we always have $w_H(s_1, s_2, \ldots, s_n) \le i$.

The case of $w_H(s_0, s_1, \ldots, s_{n-1}) > i$: Suppose $w_H(s_0, s_1, \ldots, s_{n-1}) > i + 1$ then it must be that $w_H(s_1, s_2, \ldots, s_n) > i$. In the following we assume $w_H(s_0, s_1, \ldots, s_{n-1}) = i + 1$. If now we have $s_0 = 0$, then $w_H(s_1, s_2, \ldots, s_n) \ge w_H(s_0, s_1, \ldots, s_{n-1}) > i$. If $s_0 = 1$, then $w_H(s_1, \ldots, s_{n-1}) = i$. Because $i \notin \mathrm{Ind}(h)$, we have $h(s_1, \ldots, s_{n-1}) = 0$. Hence $s_n = s_0 + h(s_1, \ldots, s_{n-1}) = 1$, which implies that $w_H(s_1, s_2, \ldots, s_n) = i + 1$. No matter which is the case, we always have $w_H(s_1, s_2, \ldots, s_n) > i$.

Therefore, we have the following result,

$$w_H(s_0, s_1, \ldots, s_{n-1}) \le i \Rightarrow w_H(s_1, s_2, \ldots, s_n) \le i, \tag{1}$$
$$w_H(s_0, s_1, \ldots, s_{n-1}) > i \Rightarrow w_H(s_1, s_2, \ldots, s_n) > i. \tag{2}$$

which directly means that either $C$ contains only states of weight $\le i$ or contains only states of weight $> i$. □

By this lemma, if there is an integer $0 \leq i \leq n-1$ that does not belong to $\mathrm{Ind}(h)$, then the cycles in $\mathrm{FSR}(x_0+h+x_n)$ are divided into two classes: the cycles that contain only states of weight $\leq i$, and the cycles that contain only states of weight $> i$. Furthermore, if there are many integers that do not belong to $\mathrm{Ind}(h)$, then the cycles in $\mathrm{FSR}(x_0+h+x_n)$ are divided into many classes, which makes the cycle structure much easier to analyze. The most simple case is $\mathrm{Ind}(h)=\emptyset$, in which the cycles are divided into $n+1$ classes with each class contains the states (these states may be located on more than one cycles) that cyclic equivalent with each other. In this simple case we have $h=0$, and we call the FSR pure circulating register.

We use $[a,b]$ to denote the set of integers between $a$ and $b$, that is, $[a,b]=\{i \mid a \leq i \leq b\}$. We allow the case of $a=b$, in which we means that $[a,a]=\{a\}$, the set of a single integer. With this notation, the index set $\mathrm{Ind}(h)$ can be uniquely written as $\mathrm{Ind}(h)=\cup_{i=1}^{u}[a_i,b_i]$, where $a_i$ and $b_i$ are integers satisfying $b_i+1<a_{i+1}$. Define $\mathrm{EInd}(h)=\cup_{i=1}^{u}[a_i,b_i+1]$, called the extended index set of $h$. It should be noticed that, $\mathrm{EInd}(h)$ is a subset of $\{0,1,\ldots,n\}$, not $\{0,1,\ldots,n-1\}$ (remember that $\mathrm{Ind}(h)$ is a subset of $\{0,1,\ldots,n-1\}$). The complementary set of $\mathrm{EInd}(h)$ is defined to be $\overline{\mathrm{EInd}(h)}=\{0,1,\ldots,n\}\setminus \mathrm{EInd}(h)$. Below is a small example that help understand these notations.

**Example 2.** *Let* $n=10$ *and* $\mathrm{Ind}(h)=\{1,2,3,5,8,9\}$*, then* $\mathrm{Ind}(h)$ *can be uniquely written as* $\mathrm{Ind}(h)=[1,3]\cup[5,5]\cup[8,9]$*. By the definition of extended index set we have* $\mathrm{EInd}(h)=[1,4]\cup[5,6]\cup[8,10]=[1,6]\cup[8,10]$*, and* $\overline{\mathrm{EInd}(h)}=\{0,7\}$*.*

Let $C$ be a cycle of $\mathrm{FSR}(x_0+h+x_n)$, and $\mathbf{S}$ be a state on $C$. Let $\mathbf{T}$ be the next state of $\mathbf{S}$. We discuss the follow two cases:

(i) If $w_H(\mathbf{S}) \in \mathrm{EInd}(h)$, then with the above notations there exists some $i$ such that $a_i \leq w_H(\mathbf{S}) \leq b_i+1$. Because $a_i-1$ and $b_i+1$ do not belong to $\mathrm{Ind}(h)$, according to the proof of Lemma 5 the state $\mathbf{T}$ has weight $w_H(\mathbf{T})>a_i-1$ and $w_H(\mathbf{T})\leq b_i+1$, so we have $a_i \leq w_H(\mathbf{T}) \leq b_i+1$, that is, $w_H(\mathbf{T})$ is located in the same interval as $w_H(\mathbf{S})$. This shows that, the weights of the states on $C$ are all located in the same interval $[a_i,b_i+1]$.

(ii) If $w_H(\mathbf{S}) \notin \mathrm{EInd}(h)$, then by the definition of $\mathrm{EInd}(h)$ we know that both $w_H(\mathbf{S})$ and $w_H(\mathbf{S})-1$ do not belong to $\mathrm{Ind}(h)$. According to the proof of Lemma 5, we have that $w_H(\mathbf{T}) \leq w_H(\mathbf{S})$ and $w_H(\mathbf{T})>w_H(\mathbf{S})-1$, which implies that $w_H(\mathbf{T})=w_H(\mathbf{S})$. This shows that, all the states on $C$ has the same weight. Actually, these states are cyclic equivalent with each other, and the cycle $C$ can be seen as a cycle in the pure circulating register.

We use $\mathcal{A}[a_i,b_i+1]$ to denote the set of cycles in $\mathrm{FSR}(x_0+h+x_n)$ whose states have weight $a_i \leq w_H(\mathbf{S}) \leq b_i+1$, and use $\mathcal{A}[r]$ to denote the set of cycles whose states have weight $w_H(\mathbf{S})=r$ ($r$ belongs to $\overline{\mathrm{EInd}(h)}$). Then the cycles of $\mathrm{FSR}(x_0+h+x_n)$ can be expressed as:

$$\mathrm{FSR}(x_0+h+x_n)=(\cup_{i=1}^{u}\mathcal{A}[a_i,b_i+1])\bigcup(\cup_{j=1}^{v}\mathcal{A}[r_j]), \tag{3}$$

with $u$ being the number of intervals in $\mathrm{EInd}(h)$ and $v$ being the size of the set $\overline{\mathrm{EInd}(h)}$. We call $\mathcal{A}[a_i,b_i+1]$ or $\mathcal{A}[r_j]$ a layer of $\mathrm{FSR}(x_0+h+x_n)$. Each layer consists of cycles whose states

either have weight that located in a particular interval $[a_i, b_i + 1]$ or have a particular weight $r_j$. It should be noted that, the layer $\mathcal{A}[a_i, b_i + 1]$ (or $\mathcal{A}[r_j]$) relies on the FSR considered, and it does not always define the same set of cycles. To be more accurate, it should be written as $\mathcal{A}[a_i, b_i + 1]_h$, but for simplicity the subscript $h$ is usually omitted. The following Figure 2 shows the layered cycle structure of a symmetric FSR.

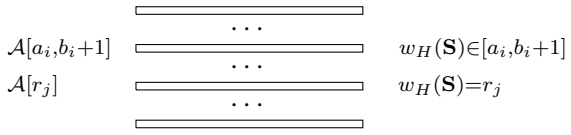


Figure 2: The layered cycle structure of a symmetric FSR

In the following we pay attention to the case that $\mathrm{FSR}(x_0 + h + x_n)$ is a scattered symmetric FSR, and give further results of their cycle structures. In the case of scattered symmetric FSRs, the extended index sets take the form of $\mathrm{EInd}(h) = \cup_{i=1}^{u}[a_i, a_i + 1]$, that is, each interval contains exactly two integers. We denote by $P$ the set of those $a_i$ which is odd and by $Q$ those which is even, then $\mathrm{EInd}(h)$ can be expressed as $\mathrm{EInd}(h) = (\cup_{p\in P}[p, p + 1]) \bigcup (\cup_{q\in Q}[q, q + 1])$. Similarly, the cycle structure of $\mathrm{FSR}(x_0 + h + x_n)$ is given by,

$$\mathrm{FSR}(x_0 + h + x_n) = (\cup_{p\in P}\mathcal{A}[p, p + 1]) \bigcup (\cup_{q\in Q}\mathcal{A}[q, q + 1]) \bigcup (\cup_{r\in\overline{\mathrm{EInd}(h)}}\mathcal{A}[r]). \tag{4}$$

Let us first consider three special scattered symmetric FSRs: the pure circulating register, the pure summing register and the complemented summing register. Their characteristic functions are $f_1 = x_0 + x_n$, $f_2 = x_0 + x_1 + \ldots + x_n$ and $f_3 = x_0 + x_1 + \ldots + x_n + 1$ respectively (the corresponding symmetric Boolean functions are, $h_1 = 0$, $h_2 = x_1 + \ldots + x_{n-1}$ and $h_3 = x_1 + \ldots + x_{n-1} + 1$). It is easily verified that, they are indeed scattered symmetric FSRs, and their corresponding index sets are $\mathrm{Ind}(h_1) = \emptyset$, $\mathrm{Ind}(h_2) = \{i \mid i \text{ is odd}, 0 \le i \le n - 1\}$ and $\mathrm{Ind}(h_3) = \{i \mid i \text{ is even}, 0 \le i \le n - 1\}$ respectively. Their layered cycle structure, according to the above formula, is given by:

$$\begin{aligned}
\mathrm{FSR}(x_0 + h_1 + x_n) &= \mathcal{A}[0] \cup \mathcal{A}[1] \cup \mathcal{A}[2] \cup \ldots \cup \mathcal{A}[n],\\
\mathrm{FSR}(x_0 + h_2 + x_n) &= \mathcal{A}[0] \cup \mathcal{A}[1, 2] \cup \mathcal{A}[3, 4] \cup \ldots,\\
\mathrm{FSR}(x_0 + h_3 + x_n) &= \mathcal{A}[0, 1] \cup \mathcal{A}[2, 3] \cup \mathcal{A}[4, 5] \cup \ldots.
\end{aligned}$$

Now we turn back to the general case of scattered symmetric FSRs. The following theorem shows that, for a general scattered symmetric FSRs, each of its layer is actually a layer of one of the three special FSRs (the pure circulating register, the pure summing register and the complemented summing register).

**Theorem 9.** *Let* $\mathrm{FSR}(x_0 + h + x_n)$ *be a scattered symmetric FSR. Its layered cycle structure is given in (4). Then for any* $r \in \overline{\mathrm{EInd}(h)}$, $p \in P$ *and* $q \in Q$ *we have,*

1. $\mathcal{A}[r]$ *is a layer of* $\mathrm{FSR}(x_0 + h_1 + x_n)$,

2. $\mathcal{A}[p, p+1]$ *is a layer of* $\mathrm{FSR}(x_0 + h_2 + x_n)$,

3. $\mathcal{A}[q, q+1]$ *is a layer of* $\mathrm{FSR}(x_0 + h_3 + x_n)$.

*Proof.* Let $C_1$, $C_2$ and $C_3$ be three cycles of $\mathrm{FSR}(x_0 + h + x_n)$, that come from the three layers $\mathcal{A}[r]$, $\mathcal{A}[p, p+1]$ and $\mathcal{A}[q, q+1]$, respectively. For the proof of this theorem we just need to show that, $C_1$, $C_2$ and $C_3$ are three cycles of $\mathrm{FSR}(x_0 + h_1 + x_n)$, $\mathrm{FSR}(x_0 + h_2 + x_n)$ and $\mathrm{FSR}(x_0 + h_3 + x_n)$, respectively.

Since the states on $C_1$ have the same weight $r$, these states are cyclic equivalent with each other. Therefore, $C_1$ is a cycle in the pure circulating register $\mathrm{FSR}(x_0 + h_1 + x_n)$.

The proofs for the other two cases are very similar, and we only prove the case of $C_2$. Let $\mathbf{S} = (s_0, s_1, \ldots, s_{n-1})$ be a state on $C_2$, and $(s_1, \ldots, s_{n-1}, s_n)$ be the next state of $\mathbf{S}$. By the recursive relation we have $s_n = s_0 + h(s_1, s_2, \ldots, s_{n-1})$. Our discussions are divided into two cases:

(i) The case of $w_H(\mathbf{S}) = p$. We further discuss whether $s_0 = 0$ or $s_0 = 1$. If $s_0 = 0$, then $w_H(s_1, s_2, \ldots, s_{n-1}) = p$. Since $p \in \mathrm{Ind}(h)$, we have $h(s_1, s_2, \ldots, s_{n-1}) = 1$, which shows that $s_n = s_0 + h(s_1, s_2, \ldots, s_{n-1}) = 1$. Therefore, $s_0 + s_1 + \ldots + s_n = 0$. If $s_0 = 1$, then $w_H(s_1, s_2, \ldots, s_{n-1}) = p - 1$. Since $p - 1 \notin \mathrm{Ind}(h)$, we have $h(s_1, s_2, \ldots, s_{n-1}) = 0$, which shows that $s_n = s_0 + h(s_1, s_2, \ldots, s_{n-1}) = 1$. Therefore, $s_0 + s_1 + \ldots + s_n = 0$.

(ii) The case of $w_H(\mathbf{S}) = p + 1$. If $s_0 = 0$, then $w_H(s_1, s_2, \ldots, s_{n-1}) = p + 1$. Since $p+1 \notin \mathrm{Ind}(h)$, we have $h(s_1, s_2, \ldots, s_{n-1}) = 0$, which shows that $s_n = s_0 + h(s_1, s_2, \ldots, s_{n-1}) = 0$. Therefore, $s_0 + s_1 + \ldots + s_n = 0$. If $s_0 = 1$, then $w_H(s_1, s_2, \ldots, s_{n-1}) = p$. Since $p \in \mathrm{Ind}(h)$, we have $h(s_1, s_2, \ldots, s_{n-1}) = 1$, which shows that $s_n = s_0 + h(s_1, s_2, \ldots, s_{n-1}) = 0$. Therefore, $s_0 + s_1 + \ldots + s_n = 0$.

No matter which is the case, we always have $s_0 + s_1 + \ldots + s_n = 0$. This implies that the states on $C_2$ satisfy the recursive relation $x_0 + x_1 + \ldots + x_n = 0$, therefore, $C_2$ can be seen as a cycle in the pure summing register $\mathrm{FSR}(x_0 + h_2 + x_n)$. □

Generally, a layer $\mathcal{A}[a, b]$ is not definitely defined because it relies on the FSR that considered, in other words, the layer $\mathcal{A}[a, b]$ in $\mathrm{FSR}(x_0 + h + x_n)$ may be different from that in $\mathrm{FSR}(x_0 + h' + x_n)$ if $h \neq h'$ (suppose they both have the layer $\mathcal{A}[a, b]$). However, for the case of scattered symmetric FSRs, the ambiguity does not exist any more. Any layer in a scattered symmetric FSR, no matter which FSR is considered, must be a layer of one of the three special FSRs. Moreover, since layers consist of cycles, the cycles in a scattered symmetric FSRs are essentially cycles of the three special FSRs. We end this subsection with an example.

**Example 3.** *Let* $h(x_1, x_2, \ldots, x_4) = E_1 + E_4$ *be a scattered symmetric Boolean function, with* $E_1$ *and* $E_4$ *being the elementary symmetric Boolean functions. Then we have* $\mathrm{EInd}(h) = [1, 2] \cup [4, 5]$, $\overline{\mathrm{EInd}(h)} = \{0, 3\}$ *and* $P = \{1\}$, $Q = \{4\}$. *The cycles of* $\mathrm{FSR}(x_0 + h + x_5)$ *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 Table 3.*

Table 2: The layers and cycles in FSR($x_0 + E_1 + E_4 + x_5$)

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

### 5.3 Improvement of the cycle joining algorithm

When applying the cycle joining algorithm (Algorithm 2) to an FSR, we can construct only one de Bruijn sequence. In this subsection, we introduce a method that can obtain a large number of de Bruijn sequences from just one (scattered symmetric) FSR.

Let FSR($x_0+h+x_n$) be a symmetric FSR. Its layered cycle structure is given in (3). Firstly, We choose a set of special states according to the following rules. For the layer $\mathcal{A}[a_i, b_i + 1]$, we choose a special state $\mathbf{S}[a_i, b_i + 1]$ such that: $w_H(\mathbf{S}[a_i, b_i + 1])) = a_i$, the least significant bit of $\mathbf{S}[a_i, b_i + 1]$ is 1, and $\mathbf{S}[a_i, b_i + 1]$ is not the cycle representative of its cycle. For the layer $\mathcal{A}[r_j]$, we choose a special state $\mathbf{S}[r_j]$ such that: the least significant bit of $\mathbf{S}[r_j]$ is 1, and $\mathbf{S}[r_j]$ is not the cycle representative of its cycle. Since the layer $\mathcal{A}[r_j]$ contains only states of weight $r_j$, the weight of $\mathbf{S}[r_j]$ must be $r_j$. Figure 3 gives an illustration on selecting special states.

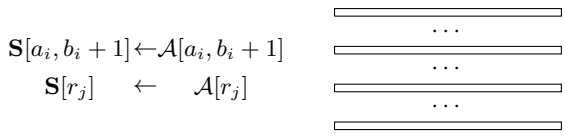


Figure 3: Selecting special states

Generally, there are many states in a layer satisfying the conditions and we can choose any one of them as the special state. But for some layers there may exists no satisfied state, in this case no special state is chosen from this layer. Therefore, the number of special states is no more than $u+v$ (because the number of layers in FSR($x_0+h+x_n$) is $u+v$). We use $V$ to denote the set of selected special states. The following lemma gives a lower bound on the number of ways to form the set $V$.

**Theorem 10.** *When $2 \leq a_i \leq n-1$, there are at least $\binom{n-2}{a_i-2}$ ways to choose a special state from the layer $\mathcal{A}[a_i, b_i + 1]$. Similarly, when $2 \leq r_j \leq n-1$, there are at least $\binom{n-2}{r_j-2}$ ways to choose a special state from the layer $\mathbf{S}[r_j]$. Especially, for any scattered symmetric FSR there are at least $2^{\frac{n-6}{2}\log n}$ ways to form the set $V$.*

*Proof.* When $2 \leq a_i \leq n-1$, $\mathbf{S}[a_i, b_i+1]$ can be any state of weight $a_i$ and of form $(*, \ldots, *, 1, 1)$.

It is easy to see that, such state is not a cycle representative, and it satisfies the conditions for special states. So there are at least $\binom{n-2}{a_i-2}$ ways to choose the special state $\mathbf{S}[a_i, b_i+1]$. Similarly, when $2 \leq r_j \leq n-1$ there are at least $\binom{n-2}{r_j-2}$ ways to choose the special state $\mathbf{S}[r_j]$.

By the above discussion, there are at least $\prod_{2\leq a_i\leq n-1}\binom{n-2}{a_i-2}\cdot\prod_{2\leq r_j\leq n-1}\binom{n-2}{r_j-2}$ ways to form the set $V$. If $\mathrm{FSR}(x_0+h+x_n)$ is a scattered symmetric FSR, then for any even number $2 \leq i \leq n-2$, at least one of the two combinatorial number $\binom{n-2}{i-2}$ and $\binom{n-2}{i-1}$ is in the product. Therefore, we have,

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

□

Each layer of $\mathrm{FSR}(x_0+h+x_n)$ includes at most one special state. So different special states belong to different layers, therefore, they belong to different cycles. Let $C_0, C_1, \ldots, C_k$ be the cycles of $\mathrm{FSR}(x_0+h+x_n)$ with $C_0$ be the cycle that containing the zero state. Then $C_0$ is in the layer $\mathcal{A}[0]$ or $\mathcal{A}[0,1]$ (depending on whether $0 \in \mathrm{Ind}(h)$ or not), and it is easy to verify that this layer contains only the cycle $C_0$. By the definition of cycle representative, $C_0$ does not have cycle representative. By the rule of selecting special state, there are no satisfied special states on $C_0$, so no special state is selected from $C_0$.

Without lose of generality, we can assume that $C_1, C_2, \ldots, C_t$ are cycles that each contains a special state (these cycles belong to different layers), and $C_{t+1}, C_{t+2}, \ldots, C_k$ are cycles that do not contain a special state. Denote the set of cycle representatives of $C_1, C_2, \ldots, C_t$ by $R_1$, and the set of cycle representatives of $C_{t+1}, C_{t+2}, \ldots, C_k$ by $R_2$, that is,

$$\begin{aligned}
R_1 &= \{\text{the cycle representative of } C_i,\ i = 1, 2, \ldots, t\},\\
R_2 &= \{\text{the cycle representative of } C_i,\ i = t+1, t+2, \ldots, k\}.
\end{aligned}$$

For any cycle of $\mathrm{FSR}(x_0+h+x_n)$ expect $C_0$, it has one state (and only one state) belongs to the set $V \cup R_2$. Furthermore, the states in the set $V \cup R_2$ are all ending with 1.

According to Theorem 2, if we interchange the predecessors of the states in $R_1 \cup R_2$ with that of their companions we get a full cycle. In order to constructing more full cycles, we substitute the set $R_1$ by the set $V$. The next theorem shows that the set $V \cup R_2$ has the same effect of joining cycles.

**Theorem 11.** *Let* $\mathrm{FSR}(x_0 + h + x_n)$ *be a symmetric FSR. The two sets* $V$ *and* $R_2$ *are defined as above. Then interchanging the predecessors of* $\mathbf{S}$ *and* $\widetilde{\mathbf{S}}$ *for all* $\mathbf{S} \in V \cup R_2$ *results in a full cycle.*

*Proof.* 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 its 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 a state ending with 1, we have $w_H(\widetilde{\mathbf{S}}_i) = w_H(\mathbf{S}_i) - 1$. If $C_i$ and $C_j$ belong to the same layer, then $\mathbf{S}_i$ must be the cycle representative of $C_i$ (because special state has the smallest weight in its layer). Therefore, the length of the longest 0-run in $C_j$ is larger than the length of the longest 0-run in $C_i$. If $C_i$ and $C_j$ belong to different layers, then there are two cases may happen: (i) $\mathbf{S}_i$ is the cycle representative of $C_i$, or (ii) $\mathbf{S}_i$ is a special state. In either case, the layer containing $C_j$ has a smaller weight than that containing $C_i$. Therefore, $T$ is a directed acyclic graph. Considering also that $T$ has $k$ edges and $k+1$ nodes (the number of edges is less than the number of nodes by one), we know that $T$ is a directed tree with root $C_0$. □

Let's do some analysis of this theorem in detail. There are two types of layers in symmetric FSRs, the type of $\mathcal{A}[a_i, b_i + 1]$ and the type of $\mathcal{A}[r_j]$. When using Theorem 2 to join cycles, the cycles in $\mathcal{A}[a_i, b_i + 1]$ are joined into some cycle either in the same layer or in a layer with a smaller weight (see the left part of Figure 4), while the cycles in the layer $\mathcal{A}[r_j]$ are all joined into some cycle that in a smaller-weight layer (see the right part of Figure 4).


Figure 4: The cycle joining process for scattered symmetric FSRs

However, if we substitute the set $R_1$ by the set $V$ and using Theorem 11 to join cycles, then the adjacency relations of cycles are changed (see the two adjacency trees defined in the proofs of Theorems 2 and 11). For layers of the type $\mathcal{A}[a_i, b_i + 1]$, since the weight of the special state $\mathbf{S}[a_i, b_i + 1]$ is $a_i$ (the minimum weight in this layer), when using this state to join cycles, the cycle must be joined into some cycle in a smaller-weight layer (if the cycle representative, not the special state, is used, then the cycle may be joined into some cycle in the same layer). This ensures that, the adjacency relations between cycles (after change) remains a directed tree, so all the cycles are eventually joined into one cycle. The same analyse is also applied to the layers of the type $\mathcal{A}[r_j]$.

In Theorem 11, we do not require that $\mathrm{FSR}(x_0 + h + x_n)$ is scattered, however, if $\mathrm{FSR}(x_0 + h + x_n)$ is not scattered then it may generate very long cycles which makes the cycle joining

algorithm extremely inefficient, because the test of representatives for long cycles takes too much time. So in the following we assume that $\mathrm{FSR}(x_0 + h + x_n)$ is a scattered symmetric FSR. Algorithm 3 gives concrete steps for generating full cycles based on scattered symmetric FSRs.

**Algorithm 3** Generation of full cycles based on scattered symmetric FSRs

INPUT: A scattered symmetric Boolean function $h$, and the initial state $\mathbf{S}_0 = (s_0, s_1, \ldots, s_{n-1})$.
OUTPUT: The full cycle $[\mathbf{S}_0, \mathbf{S}_1, \ldots, \mathbf{S}_{2^n-1}]$.
Select a special state from each layer (if can) to form the set $V$.
Calculate and store the set $R_1$.
**for** $i \in \{0, 1, \ldots, 2^n - n - 1\}$ **do**
    Define $\mathbf{S} = (s_{i+1}, \ldots, s_{i+n-1}, 1)$.
    **if** $\mathbf{S} \in V$ **then**
        $\mathbf{S}_{i+1} = (s_{i+1}, \ldots, s_{i+n-1}, s_i + h(s_{i+1}, s_{i+2}, \ldots, s_{i+n-1}) + 1)$
    **else if** $\mathbf{S}$ is not the cycle representative of its cycle **then**
        $\mathbf{S}_{i+1} = (s_{i+1}, \ldots, s_{i+n-1}, s_i + h(s_{i+1}, s_{i+2}, \ldots, s_{i+n-1}))$
    **else if** $\mathbf{S} \in R_1$ **then**
        $\mathbf{S}_{i+1} = (s_{i+1}, \ldots, s_{i+n-1}, s_i + h(s_{i+1}, s_{i+2}, \ldots, s_{i+n-1}))$
    **else**
        $\mathbf{S}_{i+1} = (s_{i+1}, \ldots, s_{i+n-1}, s_i + h(s_{i+1}, s_{i+2}, \ldots, s_{i+n-1}) + 1)$
    **end if**
**end for**

This algorithm complements the value of the feedback function if and only if the possible successor $\mathbf{S} = (s_{i+1}, \ldots, s_{i+n-1}, 1)$ belongs to the set $V \cup R_2$. According to Theorem 11, its output must be a full cycle. The following theorem analysis the efficiency of the algorithm.

**Theorem 12.** *Given a scattered symmetric FSR, it requires $\leq 2n$ state comparisons and $\leq n+1$ FSR shifts for Algorithm 3 to generate the next state from the current state.*

*Proof.* According to the rule of selecting special states, we need to select a special state from each layer (if we can). Because there are at most $n+1$ layers, the set $V$ has at most $n+1$ states. But it is easy to see that, in the case that $\mathrm{FSR}(x_0 + h + x_n)$ has exactly $n+1$ layers, the layer $\mathcal{A}[0]$ does not have special states. Therefore, the set $V$ has at most $n$ elements.

The set $R_1$ has the same size as the set $V$, so it also contains at most $n$ states. To determine the set $R_1$, we need to calculate the cycle representative for each state in $V$. This step needs $O(n^2)$ time, but it can be carried in the precomputation phase, and does not occupy the on-line running time.

The main time-consuming part of the algorithm lies in the verification of the condition of the "If" part. In this part, we have to do three judgements: (i) Determine if $\mathbf{S}$ is in $V$; (ii) Determine if $\mathbf{S}$ is the cycle representative of its cycle; and (iii) Determine if $\mathbf{S}$ is in $R_1$. Since $V$ and $R_1$ has at most $n$ elements respectively, the first and third judgements need at most $2n$

state comparisons. Because the length of the cycles in FSR($x_0 + h + x_n$) is at most $n + 1$, the second judgement needs at most $n + 1$ FSR shifts. □

We use the FSR in Example 3 to show the process of generating full cycles. We choose the set of special states to be $V = \{(01101), (10111)\}$. Then by calculation we get $R_1 = \{(10101), (11101)\}$ and $R_2 = \{(00001), (10001), (01001), (11001)\}$. The resulting full cycle is $C = [00000, 00001, 00011, 00110, 01101, 11010, 10101, 01011, 10111, 01111, 11111, 11110, 11101, 11011, 10110, 01100, 11001, 10011, 00111, 01110, 11100, 11000, 10001, 00010, 00101, 01010, 10100, 01001, 10010, 00100, 01000]$. The process of joining cycles is depicted in Figure 5.

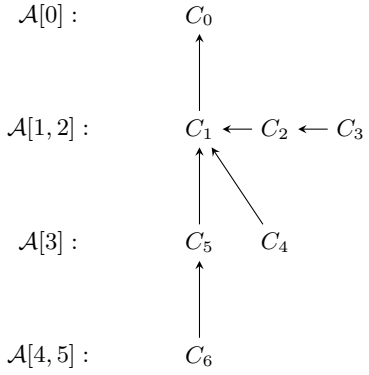


Figure 5: Join the cycles of FSR($x_0 + E_1 + E_4 + x_5$)

### 5.4 Other definition of cycle representatives

Fredricksen [6] and Etzion et al. [5] analyzed the cycle structure of the pure circulating register and the pure summing register respectively, and showed how to construct de Bruijn sequences from them. Their methods are similar to that of Jansen's, that is, define for every cycle a cycle representatives and then using representatives to form full cycles. The difference is that, their methods is applicable only to the two special FSRs, not to general ones (we note that Etzion's method is also applicable to the complemented summing register).

In this section, we want to extend their methods to the scattered symmetric FSRs. Because we have showed in Theorem 9 that a cycle of scattered symmetric FSRs is essentially a cycle of the three special FSRs (pure circulating register, pure summing register and complemented summing register), this extension is straightforward. Let's first recall some of their results.

Fredricksen [6] showed that, by properly appoint for each cycle of the pure circulating register a representative, these cycles can be joined into a full cycle by using the cycle joining method.

**Definition 2.** *[6] Let $C$ be a non-zero cycle in the pure circulating register. Regard states as binary integers and let $\mathbf{M}$ be the largest state on $C$. Suppose $|\mathbf{M}| = r \cdot 2^t$ where $r$ is an odd*

*number. Then the state whose value is $r$ is also on $C$. The cycle representative of $C$ is defined to be this state.*

By Theorem 9, there are three types of layers in scattered symmetric FSRs, $\mathcal{A}[r]$, $\mathcal{A}[p, p+1]$ and $\mathcal{A}[q, q+1]$, which correspond to the three special FSRs respectively. For the layers of the type $\mathcal{A}[r]$, its cycles are essentially pure circulating register cycles, and for these layer we can use Fredricksen's definition other than Definition 1 to define their cycle representatives. Because the cycle representatives defined by Fredricksen are always ending with 1, which has the same requirement as our definition, it can be verified that they have the same effect of joining cycles.

Etzion et al. [5] proposed a method to join the cycles of the pure summing registers. For a cycle $C = [\mathbf{S}_0, \mathbf{S}_1, \ldots, \mathbf{S}_{l-1}]$ with $\mathbf{S}_i = (s_i, s_{i+1}, \ldots, s_{i+n-1})$, its extended cycle is defined to be $C^+ = [\mathbf{S}_0^+, \mathbf{S}_1^+, \ldots, \mathbf{S}_{l-1}^+]$ with $\mathbf{S}_i^+ = (s_i, s_{i+1}, \ldots, s_{i+n-1}, s_{i+n})$. Let $C$ be a cycle of the pure summing register. Since $1 + x + \cdots + x^n \mid 1 + x^{n+1}$, the extended cycle $C^+$ is a cycle of the $n+1$ stage pure circulating register. Let $\mathbf{S}^+$ be a state on $C^+$, then the other states of $C^+$ are all cyclic equivalent with $\mathbf{S}^+$, and so they have the same weight. We define the extended weight of $C$ to be the weight of $\mathbf{S}^+$, denoted by $w_E(C)$.

Write $\mathbf{S}^+ = (s_0, s_1, \ldots, s_{n-1}, s_n)$, then we have $s_n = s_0 + s_1 + \cdots + s_{n-1}$. The extended weight of $C$ is $w_E(C) = \sum_{i=0}^{n} s_i = 2\sum_{i=0}^{n-1} s_i$. This shows that the extended weight of $C$ must be an even number. Let the extended weight of $C$ be $2k$, then it is easy to see that, the weights of the states on $C$ satisfy, $2k - 1 \leq w_H(\mathbf{S}) \leq 2k$. This gives a relation between the extended weights of cycles and the weights of their states. This relation can be expressed in another form,

$$w_E(C) = \begin{cases} w_H(\mathbf{S}) + 1 & \text{If } w_H(\mathbf{S}) \text{ is odd} \\ w_H(\mathbf{S}) & \text{If } w_H(\mathbf{S}) \text{ is even.} \end{cases}$$

If $C^+$ contains a state of the form $(\overbrace{1, \ldots, 1}^{t}, 0, \ldots, 0)$ with $t \geq 0$, then we call $C$ a run cycle; otherwise $C$ is a non-run cycle. For every non-run cycles of the pure summing registers, Etzion et al. gives a definition of cycle representative for it.

**Definition 3.** *[5] Let $C$ be a non-run cycle of the pure summing register. Then its expended cycle $C^+$ has only one state $\mathbf{S}^+$ such that: it is of the form*

$$(\overbrace{0, \ldots, 0}^{r}, \overbrace{1, \ldots, 1}^{t}, 0, *, \ldots, *, 1, 0),$$

*with (i) $r \geq 0$; (ii) $t$ is the length of the longest 1 runs; and (iii) $\mathbf{S}^+$ is the largest one of those satisfying (i) and (ii). We define the cycle representative of $C$ to be,*

$$\mathbf{S} = (\overbrace{0, \ldots, 0}^{r}, \overbrace{1, \ldots, 1}^{t}, 0, *, \ldots, *, 1).$$

Then they showed that, by using these cycle representatives the cycles with the same extended weight are joined into a single one. It is easy to see that, the set of cycles with extended

weight $w_E(C) = 2k$ is exactly the layer $\mathcal{A}[2k-1, 2k]$, so the cycles in the same layer are joined together. For joining different layers, they introduced some other method that different from the representatives-method, and we will not discuss it here. In the following, we add a simple definition of cycle representative for run cycles so that different layers can be joined together.

**Definition 4.** *Let $C$ be a (non-zero) run cycle of the pure summing register with extended weight $2k$. The cycle representative of $C$ is defined to be* $(0, \ldots, 0, \overbrace{1, \ldots, 1}^{2k-1})$.

Definitions 3 and 4 together give the full definition of cycle representative for the pure summing register, which is used in the following. It is easy to see that, the cycle representatives in the pure summing register are always ending with 1. Moreover, we can verify that: (i) for the non-run cycles, the companion of a cycle representative is located on some cycle that in the same layer; and (ii) for the run cycles, the companion of cycle representative is always located on some cycle that in a smaller-weight layer. Figure 6 gives the adjacency tree of the pure summing register.

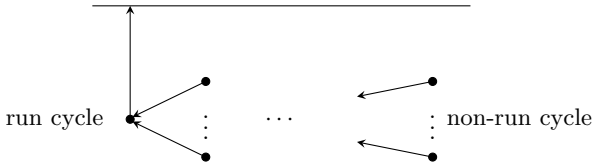


Figure 6: Adjacency tree of the pure summing register

We should note that, the method for joining cycles of pure summing registers can also apply to the complemented summing registers, with Definitions 3 and 4 unchanged. Since the layers of types $\mathcal{A}[p, p+1]$ and $\mathcal{A}[q, q+1]$ in a scattered symmetric FSR are essentially layers of the pure summing registers and the complemented summing registers. For these two types of layers, we can use Definitions 3 and 4 other than Definition 1 to define their cycle representatives. Since these definitions all require the representatives end with 1, they have the same effect of joining cycle. This gives us alternative ways to define representatives which implies more ways to generate full cycles.

# 6 Pure Circulating Registers

In searching for short-cycle FSRs, we observed that the $n$-stage pure circulating shift register, $\mathrm{FSR}(x_0 + x_n)$, is the unique $n$-stage FSR that generates cycles of length $\leq n$ only. In this subsection, a proof of this result is given.

Let $\mathbf{S} = (s_0, s_1, \ldots, s_{n-1})$ be a state of a nonsingular FSR of length $n$. Its period is defined to be the least positive integer $u \leq n$ such that $\mathbf{S} = (s_0, s_1, \ldots, s_{u-1}, \ldots, s_0, s_1, \ldots, s_{u-1})$, and

denoted by per($\mathbf{S}$). For any state, its period is a divisor of its length. The conjugate of $\mathbf{S}$ is defined to be $\widehat{\mathbf{S}} = (\bar{s}_0, s_1, \ldots, s_{n-1})$.

**Lemma 6.** *Let* $\mathbf{S}$ *be a state of length* $n$*. If* $\mathrm{per}(\mathbf{S}) < n$*, then* $\mathrm{per}(\widehat{\mathbf{S}}) = n$*.*

*Proof.* Let $u$ and $v$ be the periods of $\mathbf{S} = (s_0, s_1, \ldots, s_{n-1})$ and $\widehat{\mathbf{S}} = (\bar{s}_0, s_1, \ldots, s_{n-1})$ respectively. Since $u$ is a divisor of $n$ and $u < n$, we have $u \leq \frac{n}{2}$. Suppose $v < n$. Since $v$ is a divisor of $n$, we have $v \leq \frac{n}{2}$. It is easy to see that $u + v \neq n$, therefore, we have $u + v < n$. From $\mathrm{per}(\mathbf{S}) = u$, follows $s_{u+v} = s_v$, and from $\mathrm{per}(\widehat{\mathbf{S}}) = v$, follows $s_v = \bar{s}_0$. Therefore, $s_{u+v} = \bar{s}_0$. Similarly, from $\mathrm{per}(\widehat{\mathbf{S}}) = v$ follows $s_{u+v} = s_u$, and from $\mathrm{per}(\mathbf{S}) = u$ follows $s_u = s_0$. Hence, $s_{u+v} = s_0$, contradicting $s_{u+v} = \bar{s}_0$. □

**Corollary 2.** *Let* $C_1$ *and* $C_2$ *be two cycles in the* $n$*-stage pure circulating register. If their lengths are less than* $n$*, then* $C_1$ *and* $C_2$ *are not adjacent.*

*Proof.* Let $\mathbf{S}$ be a state on $C_1$. According to Lemma 6, $\widehat{\mathbf{S}}$ is not on $C_2$ ($\widehat{\mathbf{S}}$ is on some cycle of length $n$). This implies that $C_1$ and $C_2$ are not adjacent. □

For a state $\mathbf{S} = (s_0, s_1, \ldots, s_{n-1})$, the extend-period of $\mathbf{S}$, denoted by Eper($\mathbf{S}$), is defined to be the least positive integer $u$ such that $(s_0, s_1, \ldots, s_{n-u-1}) = (s_u, s_{u+1}, \ldots, s_{n-1})$. If there is no such integer, then we say $\mathrm{Eper}(\mathbf{S}) = n$. It can be verified that, Eper($\mathbf{S}$) is the minimum period of the periodic sequences whose first $n$ bits are $s_0 s_1 \ldots s_{n-1}$. The left shift operator $L$ is defined to be $L(s_0, s_1, \ldots, s_{n-1}) = (s_1, s_2, \ldots, s_{n-1}, s_0)$.

**Lemma 7.** *Let* $\mathbf{S} = (s_0, s_1, \ldots, s_{n-1})$ *be a state of period* $n$*, and* $\mathbf{R}$ *be the least state in the set* $\{\mathbf{S}, L(\mathbf{S}), \ldots, L^{n-1}(\mathbf{S})\}$*. Then* $\mathrm{Eper}(\mathbf{R}) = n$*.*

*Proof.* Since $\mathrm{per}(\mathbf{R}) = \mathrm{per}(\mathbf{S}) = n$, we know $\mathbf{R} \neq (0, 0, \ldots, 0)$ and $\mathbf{R} \neq (1, 1, \ldots, 1)$. There are at least one 0 and at least one 1 in $\mathbf{R}$. Let $t$ be the length of the longest run of zeros in $\mathbf{R}$. Then $\mathbf{R}$ is of the form $\mathbf{R} = (\overbrace{0, \ldots, 0}^{t}, 1, *, \ldots, *, 1)$. There may be more than one run of zeros of length $t$ in $\mathbf{R}$. Write $\mathbf{R} = (r_0, r_1, \ldots, r_{n-1})$, where $r_0 = r_1 = \cdots = r_{t-1} = 0$ and $r_{n-1} = 1$. Suppose $\mathrm{Eper}(\mathbf{R}) < n$. By the definition of extended-period, there exists some integer $0 < u < n$ such that $(r_0, r_1, \ldots, r_{n-u-1}) = (r_u, r_{u+1}, \ldots, r_{n-1})$. Since the least significant bit of $\mathbf{R}$ is 1, $r_u$ is the beginning of some run of zeros of length $t$. We claim that $u$ is not a divisor of $n$, otherwise we would have $\mathrm{per}(\mathbf{R}) = u < n$. Therefore, we have $u \neq \frac{n}{2}$. If $u < \frac{n}{2}$, since $u$ is not a divisor of $n$, we can assume $n = qu + r$, where $0 < r < u$. Define

$$v = \begin{cases} qu & \text{if } u < \frac{n}{2}, \\ u & \text{if } u > \frac{n}{2}. \end{cases}$$

It can be verified that always: (i) $\frac{n}{2} < v < n$; (ii) $r_v$ is the beginning of some run of zeros of length $t$; and (iii) $(r_0, r_1, \ldots, r_{n-v-1}) = (r_v, r_{v+1}, \ldots, r_{n-1})$. Let $m$ be the integer corresponding to $(r_0, r_1, \ldots, r_{n-v-1})$, i.e., $m = \sum_{i=0}^{n-v-1} r_i 2^{n-v-1-i}$, and $m'$ be the integer corresponding to

$(r_{n-v}, r_{n-v+1}, \ldots, r_{v-1})$. These numbers are shown in Figure 7. The rectangles in the figure denote the longest run of zeros.

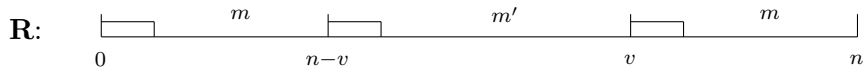


Figure 7: The numbers related to $\mathbf{R}$

Regarding states as integers, we have

$$\begin{aligned}\mathbf{R} &= m + m' \cdot 2^{n-v} + m \cdot 2^{v},\\ L^{n-v}(\mathbf{R}) &= m + m \cdot 2^{n-v} + m' \cdot 2^{2n-2v},\\ L^{v}(\mathbf{R}) &= m' + m \cdot 2^{2v-n} + m \cdot 2^{v}\end{aligned}$$

Since all three states belong to $\{\mathbf{S}, L(\mathbf{S}), \ldots, L^{n-1}(\mathbf{S})\}$ and $\mathbf{R}$ is the least state of this set, we have $L^{n-v}(\mathbf{R}) > \mathbf{R}$ and $L^{v}(\mathbf{R}) > \mathbf{R}$. However, by

$$\begin{aligned}&(L^{n-v}(\mathbf{R}) - \mathbf{R})(L^{v}(\mathbf{R}) - \mathbf{R})\\ =&[m(2^{n-v} - 2^{v}) + m'(2^{2n-2v} - 2^{n-v})][m(2^{2v-n} - 1) + m'(1 - 2^{n-v})]\\ =&-2^{n-v}[m(2^{2v-n} - 1) + m'(1 - 2^{n-v})]^2\\ <&0,\end{aligned}$$

either of $L^{n-v}(\mathbf{R})$ and $L^{v}(\mathbf{R})$ is less than $\mathbf{R}$, which is a contradiction. □

**Theorem 13.** *Let* FSR($f$) *be an $n$-stage FSR that generates only cycles of length no more than $n$, then $f = x_0 + x_n$.*

*Proof.* Let $\mathbf{S} = (s_0, s_1, \ldots, s_{n-1})$ be a state of length $n$. We call $\mathbf{S}$ a satisfied state if the next state of $\mathbf{S}$ in FSR($f$) is $L(\mathbf{S}) = (s_1, \ldots, s_{n-1}, s_0)$. For the proof of this theorem we need to show that all the states of length $n$ are satisfied states.

First consider the case $\mathrm{per}(\mathbf{S}) = n$. Let $\mathbf{R}$ be the least state in the set $\{\mathbf{S}, L(\mathbf{S}), \ldots, L^{n-1}(\mathbf{S})\}$. According to Lemma 7, $\mathrm{Eper}(\mathbf{R}) = n$. Consider the sequence $\mathbf{s}$ generated by FSR($f$) with initial state $\mathbf{R}$. Since FSR($f$) generates only cycles of length $\leq n$, we get $\mathrm{per}(\mathbf{s}) \leq n$. By the definition of extended-period, we know $\mathrm{per}(\mathbf{s}) \geq \mathrm{Eper}(\mathbf{R}) = n$. Thus, we have $\mathrm{per}(\mathbf{s}) = n$. This implies that $[\mathbf{S}, L(\mathbf{S}), \ldots, L^{n-1}(\mathbf{S})]$ is a cycle of FSR($f$) and all the states on this cycle are satisfied states. Especially, $\mathbf{S}$ is a satisfied state.

Next, consider the case $\mathrm{per}(\mathbf{S}) < n$. According to Lemma 6, $\mathrm{per}(\mathbf{S}) < n$ implies $\mathrm{per}(\widehat{\mathbf{S}}) = n$. From the above discussion, $\widehat{\mathbf{S}}$ is a satisfied state. Therefore, the next state of $\widehat{\mathbf{S}}$ in FSR($f$) is $(s_1, \ldots, s_{n-1}, \overline{s}_0)$. Since FSR($f$) is a nonsingular FSR, the successors of a conjugate pair form a companion pair. Hence, the next state of $\mathbf{S}$ in FSR($f$) is $(s_1, \ldots, s_{n-1}, s_0)$ and $\mathbf{S}$ is a satisfied state. □

# 7 Conclusion

The performance of the classical cycle joining algorithm proposed by Jansen et al. [12] was improved in this paper. Since the cycle joining algorithm needs FSRs with short cycles only, two classes of such FSRs are proposed, that are nonlinear and readily available. Based on these classes of FSRs, a large number of de Bruijn sequences can be constructed efficiently. The size of the classes of de Bruijn sequences, as well as the efficiency of the cycle joining algorithms was analyzed. Additionally, a property of the pure circulating register was proved.

# References


[1] A. H. Chan, R. A. Games, and E. L. Key, "On the complexities of De Bruijn sequences," *J. Combin. Theory, Ser. A*, vol. 33, pp. 233-246, Nov. 1982.

[2] N. Courtois and W. Meier, "Algebraic attacks on stream ciphers with linear feedback," in Advances in Cryptology-EUROCRYPT 2003 (Lecture Notes in Computer Science). Berlin, Germany: Springer-Verlag, 2003, vol. 2656, pp. 345-359.

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

[4] J. Dong and D. Pei, "Construction for de Bruijn sequences with large stage," *Proc. Designs, Codes, Cryptogr.*, vol. 85, no. 2, PP. 343-358, Nov. 2017.

[5] 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, May. 1984.

[6] H. Fredricksen, "A class of nonlinear de Bruijn cycles," *J. Comb. Theory, Ser. A*, vol. 19, no. 2, pp. 192-199, Sep. 1975.

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

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

[9] D. H. Green and K. R. Dimond, "Nonlinear product-feedback shift registers," *Proc. IEE*, vol. 117, no. 4, pp. 681-686, Apr. 1970.

[10] T. Helleseth, and T. Klove, "The number of cross-join pairs in maximum length linear sequences," *IEEE Trans. Inf. Theory*, vol. 37, no. 6, pp. 1731-1733, Nov. 1991.

[11] F. Hemmati, "A large class of nonlinear shift register sequences," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 355-359, Mar. 1982.

[12] 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, Sep. 1991.

[13] 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.

[14] 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, May. 2014.

[15] 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, Dec. 2014.

[16] C. Y. Li, X. Y. Zeng, C. L. Li, T. Helleseth, and M. Li, "Construction of de Bruijn sequences from LFSRs with reducible characteristic polynomials," *IEEE Trans. Inf. Theory*, vol. 62, no. 1, pp. 610-624, Jan. 2016.

[17] M. Li, Y. Jiang, and D. Lin, "The adjacency graphs of some feedback shift registers," *Des. Code Cryptogr.*, vol. 82, no. 3, pp. 695-713, Mar. 2017.

[18] M. Li, and D. Lin, "The adjacency graphs of LFSRs with primitive-like characteristic polynomials," *IEEE Trans. Inf. Theory*, vol. 63, no. 2, pp. 1325-1335, Feb. 2017.

[19] R. Lidl and H. Niederreiter, Finite Fields, in Encyclopedia of Mathematics and Its Applications, Reading, MA: Addison-Wesley, 1983, vol. 20.

[20] K. Mandal and G. Gong, "Cryptographically strong de Bruijn sequences with large periods," in Selected Areas in Cryptography (Lecture Notes in Computer Science), vol. 7707. Berlin, Germany: Springer-Verlag, 2013, pp. 104 – 118.

[21] K. Mandal and G. Gong, "Feedback Reconstruction and Implementations of Pseudorandom Number Generators from Composited De Bruijn Sequences," *IEEE Trans. Comput.*, Vol. 65, no. 9, pp. 2725 – 2738, Sep. 2016.

[22] T. Muir, "Note on Selected Combinations," Proc. Royal Society of Edinburgh, vol. 24, 1904.

[23] D. E. Muller, "Application of Boolean algebra to switching circuit design and to error correction," *IRE Trans. Electron. Comput.*, vol. EC-3, no. 9, pp. 6-12, Sep. 1954.

[24] 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, Nov. 1979.

[25] J. Mykkeltveit, "A proof of Golomb's conjecture for the de Bruijn graph," *J. Comb. Theory, Ser. B*, vol. 13, pp. 40-45, 1972.

[26] I. S. Reed, "A class of multiple-error-correcting codes and the decoding scheme," *IRE Trans. Inform. Theory*, vol. IT-4, pp. 38-49, Sep. 1954.

[27] J. Sawada, A. Williams and D. Wong, "A surprisingly simple de Bruijn sequence construction," *Discrete Mathematics*, vol. 339, no. 1, pp. 127-131, Jan. 2016.

[28] T. Siegenthaler, "Decrypting a class of stream ciphers using ciphertext only," *IEEE Trans. Comput.*, vol. C-34, pp. 81-85, Jan. 1985.

[29] Jan Søreng, "The periods of the sequences generated by some symmetric shift registers," *J. Comb. Theory, Ser. A*, vol. 21, no. 2, pp. 164-187, Sep. 1976.

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