Lyndon Words and de Bruijn sequences in a Subshift of Finite Type

Eduardo Moreno[*†‡]

emoreno@dim.uchile.cl

Abstract

This work generalizes the concepts of Lyndon words and de Bruijn sequence to the language of subshifts of finite type, extending its properties to this more general case. It is also studied the validity of Fredricksen and Maiorana theorem in this context and it is presented an algorithm to construct a de Bruijn sequence using Lyndon words.

1 Introduction and definitions

Let A be a finite set with a linear order $<$. A *word* on the alphabet A is a finite sequence of elements of A. The length of a word $w \in A^*$ is denoted by $|w|$. A word p is said to be a *prefix* of a word w if there exists a word u such that $w = pu$. The prefix p is proper if $p \neq w$. The definition of a *suffix* is symmetrical.

The set A^* of all the words on the alphabet A is linearly ordered by the alphabetic order induced by the order $<$ on A. By definition, $x < y$ either if x is a prefix of y or if $x = uav$, $y = ubw$ with $u, v, w \in A^*$, $a, b \in A$ and $a < b$. A basic property of the alphabetic order is the following: if $x < y$ and if x is not a prefix of y, then for all words u, v, $xu < yv$.

Two words x, y are *conjugate* if there exist words u, v such that $x = uv$ and $y = vu$. Conjugacy is an equivalence relation in A^*. A word is said to be *minimal* if it is the smallest in its conjugacy class. A word is *primitive* if it is not a proper power, i.e., if it is not of the form r^n for $r \in A^*$ and $n \geq 2$. A *Lyndon word* is a word which is both primitive and minimal.

Lyndon words were introduced under the name of *standard lexicographic sequences* in [Lyn54] and [Lyn55] to construct a basis of the free Abelian group F_n/F_{n+1}, where F_n is the nth derived group of the free group F on the set A. This free Abelian group is isomorphic to $\mathcal{L}_n(A)$, the nth homogeneous component of the free Lie algebra via the Magnus transformation. Lyndon words have

[*]Institut Gaspard Monge, Université de Marne-la-Vallée, Champs-sur-Marne, Marne-la-Vallée cedex 2, France

[†]Departamento de Ingeniería Matemática, Facultad de Ciencias Físicas y Matemáticas, Universidad de Chile, Centro de Modelamiento Matemático, UMR 2071, UCHILE-CNRS, Casilla 170-3, Correo 3, Santiago, Chile.

[‡]Partially supported by ECOS C00E03 (French-Chilean Cooperation), Proyecto MECESUP UCH0009 and CONICYT Ph.D. Fellowship.

been also utilized in very different subjects, as cryptography (see [SM90]) and stochastic differential equations (see [LL97]).

The number of Lyndon words of length n (denoted by q_n) is given by Witt's formula (see [CS01])

$$q_n = \frac{1}{n} \sum_{d|n} \mu(n/d) \left[\mathrm{Card}(A)\right]^d$$

where μ is the Möebius function. Similarly, the number of conjugacy classes (denoted by s_n), which is equal to $\sum_{d|n} q_d$, is given by

$$s_n = \frac{1}{n} \sum_{d|n} \varphi(n/d) \left[\mathrm{Card}(A)\right]^d$$

where φ is the Euler function.

The following properties are equivalent definitions of Lyndon words.

Proposition 1. *A word w is a Lyndon word if and only if for any non-empty words u, v such that $w = uv$, we have $w < vu$.*

Proposition 2. *A word $w \in A^+$ is a Lyndon word if and only if $w \in A$ or $w = lm$ with $l < m$ and l, m are Lyndon words.*

This last property gives an algorithm to construct Lyndon words. This algorithm allows the factorization of any word in A^* in non-increasing Lyndon words.

Theorem 3 (Lyndon). *Any word $w \in A^+$ may be written uniquely as a non-increasing product of Lyndon words.*

In [Duv83] is given an algorithm to produce all the words of lesser or equal length to a given integer in lexicographical order. This algorithm is efficient, it has asymptotically a constant average cost (see [BP94]).

An important combinatorial problem solved with Lyndon words is the generation of all the words of a given length. This problem has been considered in several contexts (see [Knuar] and [Reu93]). A method to do this efficiently is to use a *de Bruijn* sequence.

A de Bruijn sequence of span n is a string B_n of length $|A|^n$ such that the set of substring of length n is exactly the set of words of length n, i.e., $\{(B_n)_i \dots (B_n)_{i+n-1 \mod n} | i = 0 \dots n-1\} = \{w : w \in A^n\}$. The major problem related with de Bruijn sequences is to generate them efficiently (see [Fre82] for a survey about this). One solution to this problem, given in [FM78], is to utilize the relations existing between de Bruijn sequences and Lyndon words for obtain an efficient algorithm to produce all the words of a length n.

Theorem 4 (Fredricksen-Maiorana). *For a given n, the lexycographical concatenation of Lyndon words of length dividing n generate a de Bruijn sequence of span n.*

In this work we study and generalize these properties and theorems to any language with "forbidden" subwords, specifically, periodic words in a subshift of finite type.

Given an alphabet A, a full shift $A^{\mathbb{Z}}$ is the collection of all bi-infinite sequences of symbols from A. Let $\mathcal{F}$ be a set of words over A. We will refer to this set as the set of *forbidden blocks*. A *subshift of finite type* (SFT) S is the subset of sequences in $A^{\mathbb{Z}}$ which does not contain any block in $\mathcal{F}$.

Subshifts of finite types have been introduced by Smale [Sma73] as a basic element to understand the dynamics of smooth mappings. They play an essential role in mathematical subjects like dynamical systems, and their have provided solutions to applied problems, such as efficient coding schemes to store data on computers medias.

The *language* of a SFT S, denoted $\mathcal{L}(S)$, is the collection of words of any length into the sequences in S. A SFT is *irreducible* if for every ordered pair of blocks u, v of $\mathcal{L}(S)$ there is a block $w \in \mathcal{L}(S)$ so that uwv is a block of the language of S.

It is known that every SFT can be represented as a labeled graph, such that the language of the SFT is the label of the paths over the graph. Note that a SFT is irreducible if and only if the graph associated is strongly connected.

A bi-infinte sequence x is periodic if it is composed by infinite repetitions of a finite word w, which is called a root of x. The period of x is the length of its smallest root. In this work we will use infinite periodical sequences, represented by his smallest root. More information about these concepts can be found in [LM95].

In this article we generalize the concepts of Lyndon words and de Bruijn sequences to the language of a subshift of finite type. In particular, several equivalent definitions of Lyndon words are given. The classical theorem proving that Lyndon words form a factorization of free monoids is proved to be true with an additional hypothesis on the subshift of finite type. The relationship between Lyndon words and de Bruijn sequences is also extended to these kind of languages and we give an "efficient" algorithm to generate one of these sequences. Beyond the justification of these results by mere generalization, we hope that the introduction of Lyndon words in subshifts of finite type will allow to study the combinatorial properties of subshifts of finite type which is the aim of symbolic dynamics.

In section 2 we generalize the concept of Lyndon words to the language of a subshift of finite type and prove some basic properties. In section 3 we generalize the concept of de Bruijn cycles and we prove the existence of a de Bruijn sequence for any subshift of finite type. We also give an alternative proof of Fredricksen and Maiorana theorem to study the validity of this theorem in the general case. Finally, in section 4 we study the connection between de Bruijn sequences and Lyndon words for a subshift of finite type, and present an algorithm to produce a de Bruijn sequence using the Lyndon words of the system.

2 Lyndon words in a subshift of finite type

In this section we generalize the known results for the Lyndon words to the context of the language of a SFT. The number of Lyndon words in a SFT is also studied.

The original definition of Lyndon words needs the conjugacy and the primitively of the word. The corresponding concepts over subshift of finite type are

periodical words. Therefore, to define a Lyndon word w in a SFT it is necessary that the infinitely periodic word w^∞ is in the language of the SFT.

If S is a subshift of finite type in an alphabet A, a word w is *repeatable* in S if $w^\xi \in \mathcal{L}(S)\ \forall \xi \geq 1$. A word w is a *Lyndon word of a SFT* S if w is repeatable and if w is a Lyndon word over the alphabet without restriction. In other words, if $w^k \in \mathcal{L}(S)\ \forall k \geq 1$ and $\forall u, v \in A^+,\ l = uv \Rightarrow l < vu$. The set of Lyndon words for the SFT S will be denoted by L_S.

The following propositions are some simple generalizations of knowed properties in the original case, the first proposition is an alternative definition of a word in L_S.

Proposition 5. *A word w is in L_S if and only if it is repeatable and strictly less than any suffix, that is,*

$$w \in L_S \iff w \text{ is repeatable and if } \forall v \text{ such that } w = uv, w < v$$

Proof. Let w repeatable and $w = uv$ for $u, v \neq \varepsilon$. If $w < v$ then $w < vu$ for any prefix u, then w is a Lyndon word over A^*.

If $w \in L_S$, for any concatenation $w = uv$ we know that $w < vu$, then we only have to prove that v is not a prefix of w.

If $w = vt$ for $t \in A^+$, then $vt < vu$ and therefore $t < u$, but this means that $tv < uv = w$ and this is not possible because $w \in L_S$. □

The analogue of Proposition 2, is not possible to obtain the equivalence, because the repeatability of substrings of w cannot be assure.

Proposition 6. *If $l, m \in L_S$,with $l < m$ and lm repeatable, then $w = lm \in L_S$.*

Proof. First note that if $l < m$ then $lm < m$. If v is a suffix of m, then $m < v$ and then $lm < v$. If v' is a suffix of l, then $l < v'$. Therefore $lm < v'm$. Hence, lm is minor to any suffix, and then to L_S. □

With these properties we can study a system generated by concatenation of words in L_S. This is a particular case of a *renewal system*, which are the systems with a language of bi-infinite periodical words generated using concatenation over a finite list of words called "generating list". We define a *subshift of finite type - Lyndon factorable* (SFT-LF) as a renewal system with a generating list composed only of words in L_S.

Example 1. The *Golden Mean* is the SFT with alphabet $\{0, 1\}$ and set of forbidden subwords $\mathcal{F} = \{11\}$. This subshift of finite type is Lyndon factorable, because its language can be generated by concatenation of the words 0 and 01.

Over these subshifts, we can factorize a word (in a unique way) as a non-increasing concatenation of Lyndon words.

Proposition 7. *If S is a SFT-LF, any $w \in S$ can be written uniquely as a non-increasing product of Lyndon words of S.*

$$w = l_1 l_2 \dots l_n,\ l_i \in L_S,\ l_1 \geq l_2 \geq \dots, l_n$$

Proof. (idea) The proof is similar to the proof of the original theorem (Theorem 3), this time using the propositions 5 and 6. □

Theorem 8. *Any subshift of finite type is a factor of a subshift of finite type-Lyndon factorable.*

Proof. (idea) This result is an extension of a similar result given in [GLS91]. In this work they construct, for a given SFT S, a renewal system which has S as a factor. Extending the natural order in the original alphabet to the alphabet of the renewal system, the generating list is composed by Lyndon words. □

The number of Lyndon words is known by Witt's formula. Using the properties of subshifts of finite type we can also calculate the number of Lyndon words in our general case.

For subshifts of finite type, there is an invariant called *zeta function* which contains the number of periodic words in the language (see [RS97]). This function is defined by

$$\zeta(z) = \exp\left(\sum_{i\geq 1} \frac{\bar{p}_n}{n} z^n\right)$$

where $\bar{p}_n$ is the number of words in the language with length n. For a SFT this function can be easily calculated using the adjacency matrix M of the corresponding graph

$$\zeta(z) = \frac{1}{\det(Id - zM)}$$

and then using the Taylor's formula to obtain $\bar{p}_n$:

$$\bar{p}_n = \frac{1}{(n-1)!} \frac{d^n}{dz^n} \log \zeta(z)\Big|_{z=0}$$

Also, the words in the language of length n are exactly the power of Lyndon words of a length dividing n or its rotations, so

$$\bar{p}_n = \sum_{d|n} d \cdot \bar{q}_d$$

where $\bar{q}_d$ is the number of Lyndon words in the language of length d. Combining these equalities, we have a procedure to calculate directly the number of Lyndon words in a SFT. As before, the number $\bar{s}_n$ of conjugacy classes for words of length n in the language (and then, the number of Lyndon words of length dividing n) can be calculated using the equation

$$\bar{s}_n = \frac{1}{n} \sum_{d|n} \varphi(n/d) \bar{p}_d$$

Example 2. The *Golden Mean* has a graphical representation with two vertices and the adjacency matrix $\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$ $[\begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix}]$. Therefore, its zeta function is

$$\zeta(z) = \frac{1}{1 - z - z^2}$$

Hence, we can deduce the following values

n	1	2	3	4	5	6	7	8	9	10	11
$\bar{p}_n$	1	3	4	7	11	18	29	47	76	123	199
$\bar{q}_n$	1	1	1	1	2	2	4	5	8	11	18
$\bar{s}_n$	1	2	2	3	3	5	5	8	10	15	19

3 De Bruijn sequences in a subshift of finite type

In this section we generalize the concept of de Bruijn sequences to the language of SFTs, we prove that for any SFT exists a de Bruijn sequence (Theorem 9). Also, an alternative proof of the Fredricksen and Maiorana theorem is given, in order to extend this result to our context.

For a given subshift of finite type S and a given integer n, a de Bruijn sequence of S of span n is a string B_n^S such that the set of substring of length n $\{(B_n^S)_i \dots (B_n^S)_{i+n-1 \mod n} | i = 0 \dots n-1\}$ is exactly the set of periodic words of length n in the language of S.

We define the de Bruijn graph of span n, denoted by G_n^S, as the biggest connected component of the directed graph with $|A|^n$ vertices, labelled by the words in A^n, and the set of edges

$$E = \left\{(as, b, sb) | a, b \in A, s \in A^{n-1}, asb \in \mathcal{L}(S)\right\}$$

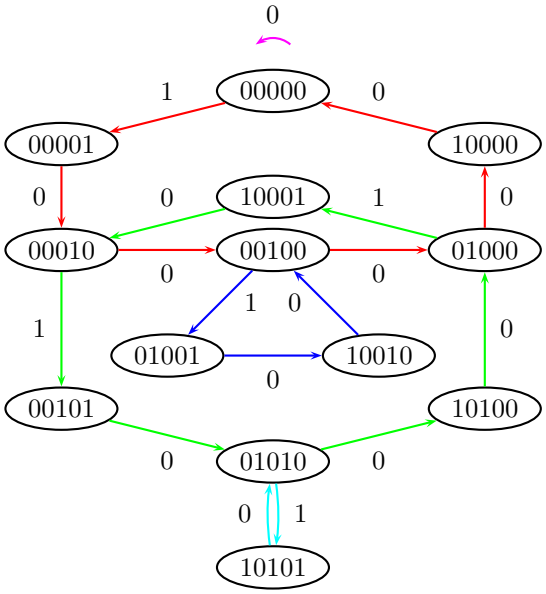

Figure 1: De Bruijn graph of span 5 for the Golden Mean

Note that a word w is a label from u to v if and only if v is a suffix of length n of uw. It is easy to see that a de Bruijn sequence of span n is exactly the label of an Eulerian cycle over G_{n-1}^S.

Theorem 9. *Every irreducible subshift of finite type S has a de Bruijn sequence B_n^S.*

Proof. We have to prove the existence of an Eulerian cycle over G_{n-1}^S, and by Euler's theorem, we only need to prove that each vertex v has the same indegree $d_-(v)$ as outdegree $d_+(v)$.

Let v a vertex of G_{n-1}^S with label $v_1 v_2 \dots v_{n-1}$, that means $d_+(v) = k$. In other words, there are $(n-1)-k$ edges producing a forbidden word. Therefore, there exist $p_l \in A$, $l = 1 \dots (n-1) - k$ such that $\exists i, j$ for which the word $v_i \dots v_{n-1} p_l v_1 \dots v_j$ is forbidden in the language.

But, in this case, the vertices with label $p_l v_1 \dots v_{n-2}$ (originally connected to v) cannot be connected to v, because any of these edges will produce a forbidden word. This is true for every l, and as every p_l is different, we conclude that $d_-(v) \leq d_+(v)$.

The other inequality can be proved using the same procedure, if a vertex label $\alpha v_1 \dots v_{n-2}$ in G^S_{n-1} has an edge to v, then v has an edge to the vertex with label $v_2 \dots v_{n-1}\alpha$ for any $\alpha \in A$, then $d_+(v) \geq d_-(v)$, proving the theorem. □

Our goal is to produce a de Bruijn sequence efficiently. The first idea is to extend the Theorem 4 to our context, but this is not always possible.

Example 3. De Bruijn sequences of the Golden Mean SFT for different lengths and the number of de Bruijn sequences.

n	p_n	de Bruijn sequence minimal	# seq.
3	4	0 001	1
4	7	0 0001 01	1
5	11	0 00001 00101	2
6	18	0 000001 001 000101 01	2
7	29	0 0000001 0001001 0000101 0010101	28
8	47	0 00000001 00001 00001001 00000101 00100101 00010101 01	216

Note that in the Golden Mean system (example 3) for a length of 6 the concatenation in lexicographical order of the Lyndon words

$$0\,000001\,000101\,001\,01$$

is not a de Bruijn Graph, in fact the words 001001, 100100, 010101 and 101010 doesn't appears, so the Fredricksen and Maiorana theorem is not true in this case. However, for the symmetrical system constructed forbidding the block 00, the concatenation of the Lyndon words is a de Bruijn sequence (and also for the system without 0^i for any i and any length). To understand this behavior, we present an alternative proof of the original Theorem 4.

Proof. (Theorem 4) Denote by a and z the minimal and the maximal letter in the alphabet A, σ the usual shift operator and B_n the de Bruijn sequence of span n.

We will prove that any conjugate $\sigma^i(w)$ of a minimal word w of length n is a factor of B_n.

Let $w = w_1 \dots w_j z^{n-j}$ be a minimal word, with $w_j < z$. First, we will show that conjugates of the form $z^{n-j-i} w_i \dots w_j z^i$ are factors of B_n, in other words, $\sigma^i(w)$ with $i = j+1 \dots n-1$.

Let m be the minimal Lyndon word with prefix $w_1 \dots w_j z^i$ (it always exists because w is a Lyndon word). Then, the previous minimal word in the lexicographical order has the form $l = l_1 \dots l_{i+j} z^{n-i-j}$ with $l_1 \dots l_{i+j} < w_1 \dots w_j z^i$. Note that if ξ is a Lyndon word of length $p < n$ with p dividing n, then $\xi^k z^{n-pk}$ is a Lyndon word for any k, so l is a Lyndon word. Hence, $z^{n-j-i} w_i \dots w_j z^i$ is a factor of lm and so is a factor of B_n.

Now we will prove this for the first j rotations. If w is not a Lyndon word (is not primitive), let $\bar{w}$ the primitive root of w and let p its length. Note that $\bar{w}$ has the form $\bar{w}_1 \dots \bar{w}_{j'} z^{p-j'}$ with $p - j' = n - j$. If $\bar{w} \neq z$, the next Lyndon word in lexicographical order x has the form $x = \bar{w}^{\frac{n}{p}-1} w_1 \dots w_{j'-1}(w_{j'}+1) b_{j'+1} \dots b_p$, so $\sigma^i(w)$ is a factor of $\bar{w}x$ for $i = 0 \dots j$. If $\bar{w} = z$ we are in the beginning case.

If w is primitive, let x be the next minimal word in lexicographical order (not necessarily primitive). So, x has the form $x_1 \dots x_{j-1}(x_j+1)b_{j+1}\dots b_n$ so $\sigma^i(w)$ is a factor of wx for $i = 0 \dots j$. If x is primitive, then wx is a factor of B_n and we are done, if not, by the previous argument, x is a prefix of $\bar{x}y$ where y is the next Lyndon word in lexicographical order, so wx is a factor of $w\bar{x}y$ and so a factor of B_n. □

Corollary 10. *Let $L = \{l_1, \dots, l_{p_n}$ be the set of Lyndon words of length n in the system without restriction, and $L_S = \{l'_1 \dots, l'_{p'_n}\}$ the set of Lyndon word of the SFT S. If $l'_1 = l_j$, $l'_2 = l_{j+1}$, …, $l'_{p'_n} = l_{p_n}$ then the lexicographical concatenation of the words in L_S is a de Bruijn sequence.*

Proof. (idea) As we see in previous proof, the rotations of a minimal word not beginning with z are factors of the concatenation of the next Lyndon words in the lexicographical order. The rotations beginning with z are a factor of the concatenation of the Lyndon word with the previous Lyndon word in the order, so we only have to assure this for the first word in L_S, but this is true because the word z^n is a suffix of the concatenation of L_S. □

4 Constructing a de Bruijn sequence in a SFT

As we have seen, the easy construction of de Bruijn sequences given by theorem 4 is not always true in our context, so in this section we construct an algorithm to generate a de Bruijn sequence. This algorithm will use the relations between the de Bruijn sequences, the de Bruijn graph and the Lyndon words of the system.

Theorem 11. *Let G^S_{n-1} be the de Bruijn graph of span $n-1$ for the language of the SFT S. For this graph, the cycles of a length dividing n are a partition of it edges.*

Proof. We prove that any edge of the graph has one and only one cycle of length dividing n.

Let $\vec{e}$ be an edge from the vertex with label au to the vertex with label ub with $a, b \in A$ (then, the label of $\vec{e}$ is b). By the construction of the graph, there is a directed path of length $n-1$ from vertex ub to vertex au with label au. Therefore, the union of this path with the edge $\vec{e}$ form a directed path of length n with label aub corresponding to one or more repetitions of a cycle of a length dividing n, proving the existence of one cycle.

Let us suppose now that there are two cycles of a length dividing n using the edge $\vec{e}$. For each cycle we can repeat the path over the graph constructing two paths C and C' of length n. Note that these paths must have the same label.

Starting from $\vec{e}$, we can backtrack the edges in the cycles until the cycles separate. Let $\vec{f}$ and $\vec{f'}$ be the edges of this separation and let assume that the label of these edges is β. Note that the arriving vertex of $\vec{f}$ and $\vec{f'}$ is the same. If this vertex has label $u\beta$ with $u \in A^{n-2}$ then $\vec{f}$ start in a vertex with label $u\alpha$ and $\vec{f'}$ in a vertex with label $u\alpha'$ with $\alpha, \alpha' \in A$.

Then, the cycle C has the label $u\beta\alpha$ and the cycle C' has the label $u\beta\alpha'$. But this means that $\alpha = \alpha'$, producing a contradiction because there are not two vertices with the same label. This proves the unity of the cycles. □

Corollary 12. *The set of Lyndon words of length dividing n of a SFT S corresponds to a partition of edges over G^S_{n-1}.*

Proof. (idea) Any cycle of length dividing n can be associated with his minimal label, so we have to prove that the label is primitive. With that we have an injection between the cycles of length dividing n and the Lyndon words of the system. By cardinality we prove that any Lyndon word have an associated cycle, finishing the proof. □

Note that any cycle can be constructed as a sum of cycles in this partition, therefore, any de Bruijn sequence (an Eulerian cycle over G^S_{n-1}) can be constructed using Lyndon words.

Over the graph, we can start with any edge and follow the corresponding cycle in the partition, until we reach to an intersection with other cycle in the partition. At this point we can follow the other cycle and when we return to the intersection we continue with the original cycle. Using this procedure iteratively we can construct an Eulerian cycle.

This idea produces an algorithm to generate a de Bruijn sequence: starting with a string u equal to an arbitrary Lyndon word, and iteratively checking if the substring $u_{i-n-1} \dots u_i \overline{u_{i+1}}$ is a power of a conjugation of a Lyndon word not yet included, where $\bar{\alpha}$ is the successor of α in A. If this is the case, the string $\overline{u_{i+1}} u_{i-n-1} \dots u_i$ is inserted in u. This algorithm finishes with a de Bruijn sequence. Note that this algorithm needs (only) the list of Lyndon words, but this can be done using the same procedure utilized in the unrestricted case, eliminating the Lyndon words that are not in the language.

ALGORITHM:
INPUT:
    n: Length of the words.
    $L_S = \{L^i\}$: Lyndon words of S with length dividing n.
begin
    $SizeLeng \leftarrow$ Size of the language of S.
    $u \leftarrow$ a word of L_S.
    $L_S \leftarrow L_S \setminus \{u\}$.
    for $i = 1$ **to** $SizeLeng$
        if $u_{i-n-1} \dots u_i \overline{u_{i+1}} = (\sigma^k(L^j))^l$ for a k, l
            $u \leftarrow u_1 \dots u_i \sigma^{k-1}(L^j) u_{i+1} \dots$
            $L_S \leftarrow L_S \setminus L^j$
        endif
    endfor
end

This algorithm can be implemented in $\mathcal{O}(n \log n)$ using a prefix tree for search the forbidden words (see [GBY91]).

Note that the algorithm 4 in [Fre82] is a particular case of this algorithm using the language without forbidden words.

Example 4. Example of the algorithm for the Golden Mean system with length $n = 6$ and Lyndon words $L_S = \{0, 000001, 000101, 001, 01\}$. See Figure 1 for a graphical interpretation of the algorithm.

000001	We start with the first word in L_S.
$\downarrow$ 000001	In this case 000011 is not in L_S.
0 $\downarrow$ 00001	here 000101 is in L_S, we include (100010)
0(100010)00001	
01 $\downarrow$ 0001000001	here 001011 is not in L_S
010 $\downarrow$ 001000001	here 010101 is in L_S, we include (10)
0(10(10)0010)00001	
0101 $\downarrow$ 0001000001	101011 is not in L_S
01010 $\downarrow$ 001000001	010101 was already included
010100 $\downarrow$ 01000001	101001 is not in L_S
0101000 $\downarrow$ 1000001	010000 was already included
0101000 $\downarrow$ 1000001	100011 is not in L_S
01010001 $\downarrow$ 000001	000101 was already included
010100010 $\downarrow$ 00001	001001 is not included, we include (100)
0101000100 $\downarrow$ 0001	
0(10(10)0010)0(100)0001	010011 is not in L_S
01010001001 $\downarrow$ 000001	100101 is not in L_S
010100010010 $\downarrow$ 00001	001001 was already included
0101000100100 $\downarrow$ 0001	010001 was already included
01010001001000 $\downarrow$ 001	100001 is not in L_S
010100010010000 $\downarrow$ 01	000000 is not included, we include (0)
0101000100100000 $\downarrow$ 1	
0(10(10)0010)0(100)000(0)1	000000 was already included
01010001001000000 $\downarrow$ 1	end.
010100010010000001 $\downarrow$	is a de Bruijn sequence
0101000100100000001	

Acknowledgement: The author wishes to thank Dominique Perrin for his helpful comments.

References

[BP94] J. Berstel and M. Pocchiola, *Average cost of duval's algorithm for generating lyndon words*, Theorical Computer Science **132** (1994), 415–425.

[CS01] H. Crapo and D. Senato (eds.), *Algebraic combinatorics and computer science*, ch. Enumerative combinatorics on words, Springer Verlag, 2001.

[Duv83] Jean-Pierre Duval, *Factorizing words over an ordered alphabet.*, J. Algorithms **4** (1983), 363–381.

[FM78] Harold Fredricksen and James Maiorana, *Necklaces of beads in k colors and k-ary de bruijn sequences*, Discrete Mathematics **23** (1978), 207–210.

[Fre82] Harold Fredricksen, *A survey of full length nonlinear shift register cycle algorithms*, SIAM Review **24** (1982), no. 2, 195–221.

[GBY91] G.H. Gonnet and R. Baeza-Yates, *Handbook of algorithms and data structures*, second ed., Addison Wesley, 1991.

[GLS91] Jacob Goldberger, Douglas Lind, and Meir Smorodinsky, *The entropies of renewal systems*, Israel Journal of Mathematics **33** (1991), 1–23.

[Knuar] Donald E. Knuth, *The art of computer programming*, vol. 4, Addison Wesley, to appear.

[LL97] C. W. Li and X. Q. Liu, *Approximation of multiple stochastic integrals and its application to stochastic differential equations*, Nonlinear Analysis **30** (1997), no. 2, 697–708.

[LM95] Douglas Lind and Brian Marcus, *Symbolic dynamics and codings*, Cambridge University Press, 1995.

[Lyn54] R.C. Lyndon, *On burnside problem i*, Trans. Am. Math. Soc. **77** (1954), 202–215.

[Lyn55] ———, *On burnside problem ii*, Trans. Am. Math. Soc. **78** (1955), 329–332.

[Reu93] Christophe Reutenauer, *Free lie algebras*, The Clarendon Press Oxford University Press, New York, 1993.

[RS97] G. Rozenberg and A. Salomaa (eds.), *Handbook of formal languages*, vol. 2, ch. 10, Springer Verlag, 1997.

[SM90] Rani Siromoney and Lisa Mathew, *A public key cryptosystem based on lyndon words*, Information Processing Letters **35** (1990), no. 1, 33–36.

[Sma73] S. Smale, *Differentiable dynamical systems*, Bull Amer. Math. Soc. (1973), 747–817.