Minimal de Bruijn Sequence in a Language with Forbidden Substrings*

Eduardo Moreno[1,2] and Martín Matamala[1]

[1] 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.
emoreno@dim.uchile.cl, mmatamal@dim.uchile.cl

[2] Institut Gaspard Monge, Université de Marne-la-Vallée, Champs-sur-Marne, 77454 Marne-la-Vallée cedex 2, France.

Abstract. In this work we give an algorithm to construct the minimal de Bruijn sequence of span n in a language with forbidden substrings. This algorithm is based in the structure of the de Bruijn graph of the language, and uses the BEST Theorem to construct a rooted tree over this graph in order to obtain the Eulerian cycle of minimal label between all Eulerian cycles of the graph.

1 Introduction

Given a language, a de Bruijn sequence of span n is a periodic sequence such that every n-tuple in the language (and no other n-tuple) occurs exactly once. Its first knowed description appears as a Sanskrit word *yamátárájabhánasalagám* which was a memory aid for Indian drummers, where the accented/unaccented syllables represent long/shorts beats, so all possible triplets of short and long beats are included in the word. De Bruijn sequences are also knowed as "shift register sequences" and was originally studied by N. G. De Bruijn for the binary alphabet [1]. These sequences have many different applications, such as memory wheels in computers and other technological device, network models, DNA algorithms, pseudo-random number generation, modern public-key cryptographic schemes, to mention a few (see [2],[3],[4]). Historically, de Bruijn sequence was studied in an arbitrary alphabet considering the language of all the n-tuples. There is a big number of de Bruijn sequence in this case, but only a few can be generated efficiently, see [5] for a survey about this subject. In 1978, Fredricksen and Maiorana [6] give an algorithm to generate a de Bruijn sequence of span n based in the Lyndon words of the language, which resulted to be the minimal one in the lexicographic order, and this algorithm was proved to be efficient [7]. Recently, the study of these concepts was extended to languages with forbidden substrings: in [8] was given efficient algorithms to generate all the words in a language with one forbidden substring, in [9] the concept of de Bruijn sequences

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

was generalized to restricted languages with a finite set of forbidden substrings and it was proved the existence of these sequences and presented an algorithm to generate one of them, however, to find the minimal sequence is a non-trivial problem in this more general case. This problem is closely related to the "shortest common superstring problem" which is a important problem in the areas of DNA sequencing and data compression.

In this work we give an algorithm to generate the minimal de Bruijn sequence in a language with forbidden substrings, using the structure of the de Bruijn graph of the language and applying the BEST Theorem to find an Eulerian cycle over the graph with the minimal label.

In section 2 we present some definitions and previous results on de Bruijn sequences and the BEST Theorem, necessary to understand the main problem, and we prove a result related with the BEST Theorem which will be useful in the following sections. In section 3 we study the main problem, giving some results on the structure of the de Bruijn graph to obtain an algorithm to produce the minimal de Bruijn sequence. Finally, in section 4 we present some remarks and extensions to this work, and comments on the implementation and complexity of the algorithm.

2 De Bruijn Sequence of Restricted Languages

2.1 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, whose length is denoted by $|w|$.

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 any pair of words u, v, $xu < yv$.

A word p is said to be a *factor* of a word w if there exists words $u, v \in A^*$ such that $w = upv$. If v is the empty word ϵ then p is called a *prefix* of w, and if u is empty then is called a *suffix* of w. If $p \neq w$ then p is a *proper factor*, *proper prefix* or *proper suffix* respectively.

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

Given a set $\mathcal{F}$ of forbidden blocks, a word w is in the language if the periodical word w^∞, composed by infinite repetitions of w, is in the language of the SFT defined by $\mathcal{F}$. The set of all the words of length n in the language defined by $\mathcal{F}$ will be denoted by $\mathcal{W}_k^{\mathcal{F}}(n)$, where k is the size of the alphabet A.

A SFT is *irreducible* if for every ordered pair of blocks u, v in the language there is a block w in the language so that uwv is a block of the language.

A de Bruijn sequence of span n in a restricted language is a string $B^{\mathcal{F},n}$ of length $\left|\mathcal{W}_k^{\mathcal{F}}(n)\right|$ such that all the words in the language of length n are factors of $B^{\mathcal{F},n}$. In other words,

$$\{(B^{\mathcal{F},n})_i \dots (B^{\mathcal{F},n})_{i+n-1 \bmod n} | i = 0 \dots n-1\} = \mathcal{W}_k^{\mathcal{F}}(n)$$

These concepts was studied in [9], extending the known results on subshifts of finite type to this context. In particular two results are relevant in this work, the first one is a bound in the number of words of length n in the language:

$$\left|\mathcal{W}_k^{\mathcal{F}}(n)\right| = \Theta\left(\lambda^n\right)$$

where $\log(\lambda)$ is the *entropy* of the system (see [10]). The second result prove the existence of a de Bruijn sequence:

Theorem 1. *For any set of forbidden substrings $\mathcal{F}$ defining an irreducible subshift of finite type, there exists a de Bruijn sequence of span n.*

This last theorem is a direct consequence of the fact that the de Bruijn graph of span n is an Eulerian graph. The *de Bruijn graph* of span n, denoted by $G^{\mathcal{F},n}$, is the biggest connected component of the directed graph with $|A|^n$ vertices, labelled by the words in A^n, and the set of arcs

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

Note that if the SFT is irreducible, this graph has only one connected component of size greater than 1, so there is no ambiguity in the definition.

There exists a bijection between the arcs of $G^{\mathcal{F},n}$ and the words in $\mathcal{W}_k^{\mathcal{F}}(n+1)$, because to each arc with label $a \in A$ with tail at a vertex with label $s \in A^n$ we can associate the word sa which is, by definition, a word in $\mathcal{W}_k^{\mathcal{F}}(n+1)$. Also, a word w is a label from u to v if and only if v is a suffix of length n of uw. With these two properties it is easy to see that a de Bruijn sequence of span $n+1$ is exactly the label of an Eulerian cycle over $G^{\mathcal{F},n}$.

2.2 The BEST Theorem

BEST is an acronym of N. G. de Bruijn, T. van Aardenne-Ehrenfest, C. A. B. Smith and W. T. Tutte, the BEST Theorem (see [11]) gives a correspondence between Eulerian cycles in a digraph and its rooted trees converging to the root vertex.

Let r be a vertex of an Eulerian digraph $G = (V, E)$, a spanning tree converging to the root r is a spanning tree such that there exists a directed path from all vertices to the root.

Given an Eulerian cycle starting at the root of an Eulerian digraph, if for every vertex of G we take the last arc with tail at this vertex in the cycle then we obtain a spanning tree converging to the root. Conversely, given a spanning tree converging to the root, if we start a path over G starting at the root and

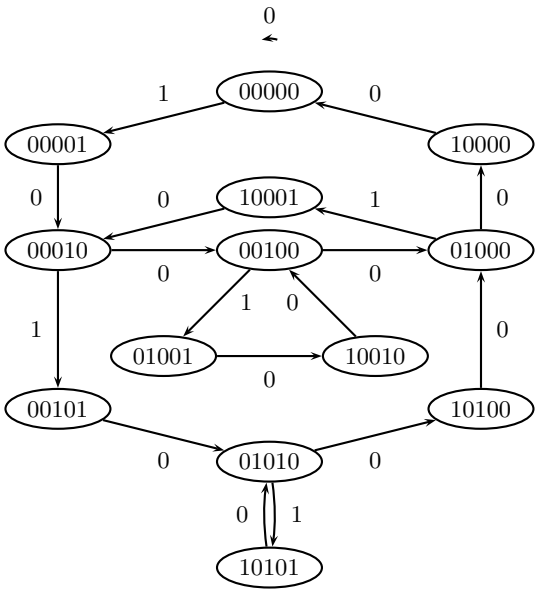

Fig. 1. De Bruijn digraph of span 5 for the Golden Mean ($\mathcal{F} = \{11\}$)

going always through an unvisited arc, such that at every vertex we use the arc in the tree only if all the others arcs with tail at this vertex was already visited, then we obtain an Eulerian cycle. A walk over the graph of this kind will be called a walk "avoiding the tree".

The BEST Theorem proves that for every different spanning tree we have a different Eulerian cycle. Therefore it allows also to calculate the exact number of Eulerian cycles on a digraph, which is given by

$$C_{\mathcal{F}} = M_T \cdot \prod_{i=1}^{|V|} (d^+(v_i) - 1)!$$

where M_T is the number of rooted spanning trees converging to a given vertex. We bound the second term by $((\bar{d}^+ - 1)!)^{|V|}$ where $\bar{d}^+$ is the mean of the outgoing degrees over all the vertices, so we have a lower bound to the number of de Bruijn sequences

$$C_{\mathcal{F}} = \Omega\left(\lfloor \lambda - 1 \rfloor !^{\lambda^{n-1}}\right)$$

in particular, for a system with $\lambda \geq 3$ the number of the Bruijn sequences of span n is exponential in the number of words in the language of length $n-1$. In the systems with $3 > \lambda > 1$ this bound is generally also true, because the underestimated term M_T is generally exponential, for example, in the system without restrictions of alphabet $\{0, 1\}$, this term is equal to $2^{2^{n-1}}$.

We will say that a walk over the graph *exhausts* a vertex if the walk used all the arc having the vertex as head or tail. The next lemma study in which order the vertices are exhausted in a Eulerian cycle defined by a walk avoiding a spanning tree T converging to a root. This is a general lemma and will be used implicitly in all this work, so it is presented in this section.

Given a tree T, we will denote T_u the component of $T \setminus e_v$ containing v, where e_v is the arc of T with tail at v, this component will be called "subtree of v".

Lemma 2. *Let W be a walk avoiding T, let v be a vertex and let Wv the subpath of W starting at the same vertex and finishing when it exhausts the vertex v. Then for each vertex u in T_v, u is exhausted in Wv.*

Proof. By induction in the depth of the subtree with root v. If v is a leaf of T then v do not have vertices in T_v. If v is not a leaf, applying induction hypothesis to all the sons of v we prove the result. □

3 Minimal de Bruijn Sequence

In this section, we construct a spanning tree converging to a particular root vertex to obtain the Eulerian cycle of minimum label over all the Eulerian cycles starting at the root vertex. This is a difficult problem because generally there exists an exponential number of Eulerian cycles.

In the unrestricted case, if we start at the vertex with the biggest label, and we follow at every vertex the arc with the lowest label between the unvisited arcs, we obtain the Eulerian cycle with the minimal label. This cycle has associated the tree composed by the arcs with labels equal to the biggest letter in the alphabet, which is effectively a tree with root at the vertex of maximal label.

In the restricted case, if we repeat this strategy and we follow the unvisited arc with the lowest label, we do not obtain necessarily an Eulerian cycle, because the graph composed by arcs with maximal label on every vertex is not necessarily a spanning tree converging to the root due to the existence of cycles.

The idea of the algorithm is to construct an spanning tree converging to the root, such that at each vertex the arc in the tree is the arc with the biggest label possible. To construct this tree, we will choose the vertex with the maximal label as root, and we will include the outgoing arc of maximum label at each non-root vertex. The main theorem of this section characterize the cycles in our construction, so we will fix these cycles and will obtain the desired tree, and therefore we will have the Eurelian cycle of minimum label between all Eulerian cycles starting at this root vertex.

More detailed, let $m = m_1, \dots m_n$ be the label of the vertex of $G^{\mathcal{F},n}$ of maximum label in the lexicographic order. Let T be the subgraph of $G^{\mathcal{F},n}$ composed by the same set of vertex and for each non-root vertex, the arc with tail at the vertex with maximal label, where the root vertex is the vertex of label m.

If there is no cycle in T then it is a tree and we are done: we start at the root and at each vertex we go through the arc of minimum label between the unvisited

arc with tail at this vertex: we will finish with a Eulerian cycle of minimum label. If exists a cycle in T, the previous strategy will not be an Eulerian cycle, because all arcs in cycles will not be visited, so we will modify the graph T to obtain a tree.

First of all, we will prove some properties of the de Bruijn graph to understand the structure of the arcs and cycles in T.

Let be a vertex of label u, we define $g(u)$ as the length of the maximal suffix of u which is a prefix of m:

$$g(u) = \max\{i : u_{n-i+1} \dots u_n = m_1 \dots m_i\}$$

Note that in the unrestricted case, $g(u)$ is the distance over the graph from the vertex with label u to the vertex of maximal label. This function will be essential in the study of T. The next lemma give us a bound over the label of the arc in T in term of the function $g(\cdot)$.

Lemma 3. *If a non-root vertex has label u then the arcs with tail at this vertex have labels $\alpha \leq m_{g(u)+1}$.*

Proof. Suppose $g(u) = i$, if exists an arc with tail at the vertex with label $\alpha > m_{i+1}$ then the word $u\alpha$ is in the language, so the word $m_1 \dots m_i \alpha u_1 \dots u_{n-i}$ is in the language and there exists a vertex with label $m_1 \dots m_i \alpha u_1 \dots u_{n-i-1}$ and an arc with label u_{n-i} with tail at this last vertex, which is a contradiction with the maximality of the label m. □

The next lemma proves that at every non-root vertex u, only the arc of label $w_{g(u)+1}$ does not go to a vertex v with $g(v) = 0$.

Lemma 4. *Let be an arc of label α going from a vertex of label u to a vertex of label v. If $\alpha < w_{g(u)+1}$ then $g(v) = 0$.*

Proof. Suppose $g(u) = i$, that means $u_{n-i+1} = m_1$, …, $u_n = m_i$. If $g(v) > g(u)$ then $g(v) = g(u)+1$ that means $v_{n-i} \dots v_n = m_1 \dots m_{i+1}$ but this is not possible because $v_n = \alpha \neq m_{i+1}$. If $g(u) = g(v) > 0$ then $u_{n-i+1} = v_{n-i+1} = m_1$, …, $u_n = v_n = m_i$ but $v_j = u_{j+1}$ for $j = 1 \dots n-1$ so $m_1 = m_2 = \dots = m_n = \alpha$, but $\alpha < m_{i+1}$ and then m is not maximal ($m_{i+1} \dots m_n m_1 \dots m_i$ is a label greater than m). Finally if $g(u) > g(v) > 0$, we know that $m_i = u_{n-g(u)+i} = v_{n-g(v)+i}$ and $u_{i+1} = v_i$, so $m_i = v_{n-g(v)+i} = u_{n-g(v)+i+1} = m_{g(u)-g(v)+i+1}$ for $i = 1 \dots g(v)-1$, and also $v_n = m_{g(v)} = \alpha$, but $m_{g(u)+1} > \alpha$ therefore $m_{g(u)-g(v)+1} \dots m_{g(u)+1}$ is a factor of m lexicographical greater than $m_1 \dots m_{g(v)}$, so m is not the maximal label. □

Note that in the unrestricted case, all the arcs not in the tree go either from a non-leaf vertex to a leaf or from a leaf to another leaf.

For a vertex with label u, we will call it a *floor* vertex if $g(u) = 0$, and a *restricted* vertex if the arc with tail at this vertex with maximum label has label $\alpha < w_{g(u)+1}$. Evidently, an arc going from a restricted vertex will go to a floor vertex, so we have the next corollary.

Corollary 5. *If a cycle in T contains l restricted vertices, then it has l floor vertices.*

We will use these lemmas to characterize the labels of the vertices in cycles.

Theorem 6. *If a cycle in T contains l restricted vertices numbered in the order of the cycle and with labels $u^1, \dots u^l$ respectively, and the arc of maximum label with tail at u^j has label α^j for $j = 1 \dots l$, then $\sum_j \left(g(u^j) + 1\right)$ divides $n+1$ and the labels of the vertices in the cycle are factors of length n of the periodic word $\left(w_1 \dots w_{g(u^1)}\alpha^1 w_1 \dots w_{g(u^2)}\alpha^2 \dots w_1 \dots w_{g(u^l)}\alpha^l\right)^\infty$. Moreover,*

$$u^j = w_1 \dots w_{g_{j+1}}\alpha^{j+1} w_1 \dots w_{g_{j+2}}\alpha^{j+2} \dots\dots \alpha^{j-1} w_1 \dots w_{g_j}$$

Proof. For simplicity we will denote $g(u^j)$ as g_j. Let i be the distance between the next vertex in the cycle after u^{j-1} and the vertex u^j. By Lemma 4, $g(u^j) = i$, in other word $u^j = \dots \alpha^{j-1} w_1 \dots w_{g(u^j)}$.

This is true for any j, so the labels of the vertices in the cycle are factors of length n of the periodic word

$$\left(w_1 \dots w_{g(u^1)}\alpha^1 w_1 \dots w_{g(u^2)}\alpha^2 \dots w_1 \dots w_{g(u^l)}\alpha^l\right)^\infty \tag{1}$$

and the label of u^j is the factor finishing in $w_1 \dots w_{g(u^j)}$.

If $u^j = u_1 \dots u_n$, then $u^{j+1} = u_{g_{j+1}+2} \dots u_n \alpha^j w_1 \dots w_{g_{j+1}}$. From the vertex u^{j+1}, which is a *restricted* vertex, we follow in the cycle by the arc with label α^{j+1}, and this is the arc with maximum label between the arc with tail at u^{j+1}, so the word $u_{g_{j+1}+2} \dots u_n \alpha^j w_1 \dots w_{g_{j+1}} a$ is forbidden $\forall a > \alpha^{i+1}$. Since the previous vertex in the cycle has label $u_{g_{j+1}+1} \dots u_n \alpha^j w_1 \dots w_{g_{j+1}-1}$ and continue with an arc of label $w_{g_{j+1}}$, to satisfy the previous condition we conclude that $u_{g_{j+1}+1} \leq \alpha^{j+1}$.

But this last vertex continues in the cycle with its maximum label $w_{g_{j+1}}$, and the previous vertex in the cycle has label $u_{g_{j+1}} \dots u_n \alpha^j w_1 \dots w_{g_{j+1}-2}$, so $u_{g_{j+1}} \leq w_{g_{j+1}}$. Repeating this argument we can prove that for any j, $(u^j)_k \leq w_k$ $\forall k \leq g(u^{j+1})$ and $(u^j)_{g_{j+1}+1} \leq \alpha^{j+1}$, so the only factor of length n of the word in (1) fulfilling these restrictions is the one fulfilling the inequalities as equalities, i.e. $\forall j$ $(u^j)_k = w_k$ $\forall k \leq g(u^{j+1})$ and $(u^j)_{g_{j+1}+1} = \alpha^{j+1}$. In other words, the vertex u^j has label

$$w_1 \dots w_{g_{j+1}}\alpha^{j+1} w_1 \dots w_{g_{j+2}}\alpha^{j+2} \dots\dots \alpha^{j-1} w_1 \dots w_{g_j}$$

in particular, $\sum_k (g(u^k) + 1)$ divide $n+1$. □

Corollary 7. *There are no cycles in $G \setminus T$ composed only by vertices of cycles in T.*

Proof. By Lemma 4, cycles in $G \setminus T$ are composed by *floor* vertices, so if exists a cycle C in $G \setminus T$ between vertices of cycles in T then the vertices of C are *floor* vertices. However these vertices are also vertices of cycles in T, hence by

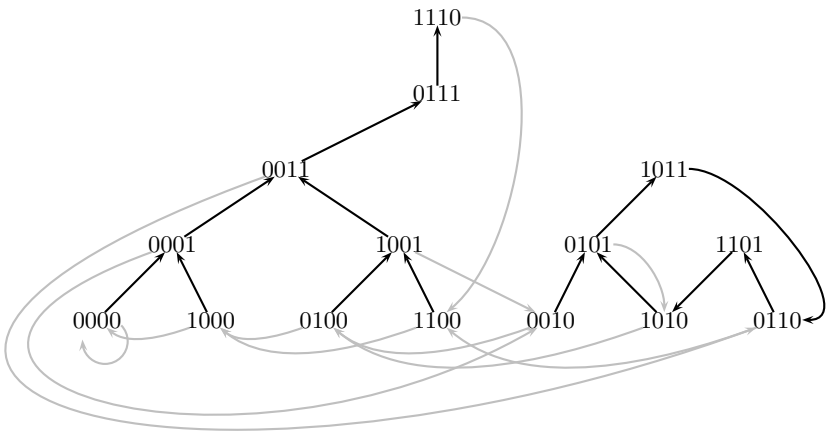

Fig. 2. Example of the graph T for $n = 4$ and $\mathcal{F} = \{01111\}$ in a binary alphabet.

Theorem 6 they have a label finishing by α^i for some i, and therefore the arcs in C have labels α^i for some i, so the labels of vertices in C are composed by a sequence of α^i for different i. Nevertheless, a vertex in a cycle of T contains w_1 in its label, and so $w_1 = \alpha^i$ for some i, which is not possible.

Now we will give an algorithm to do a walk over the graph, without repeating arcs, with the minimal label possible. We will modify T such that at every vertex the arc in T will be the last arc visited by the path between the arcs with tail at this vertex, and we will finish with a tree $\bar{T}$, proving that the walk is an Eulerian path of minimal label.

Procedure. *We start at the root vertex and we continue choosing at each step the unvisited arc of minimum label between all the unvisited arcs with tail at the current vertex. If we arrive to a vertex in a cycle C of T such that it has two unvisited arcs going out and all the others vertex of C only have the arcs in C unvisited, then we continue the path by the arc in C, and we continue with the original strategy.*

Theorem 8. *The previous procedure finishes with an Eulerian cycle of $G^{\mathcal{F},n}$ of minimal label between all labels of Eulerian cycles starting at the root.*

Proof. Suppose that we arrive to a vertex in a cycle C of T such that it has two unvisited arcs going out, one on them in the cycle (the one of maximum label) and all the others vertex of C only have the arc in C unvisited. If we continue by the arc of minimum label then the path will not uses the arcs of C because we do not have an unvisited arc arriving to a vertex of C. In other words, at this point all paths choosing the arc of minimal label instead of the arc in C will not visit the arcs in C and therefore they will not be Eulerian paths. We continue the path by the arc of C, and we modify T removing the arc in C with

tail at this vertex, and adding the other arc with tail at this vertex. After that, the path goes trough the cycle C visiting all arcs of C and continue by the new arc included in T. Note that this path will be the path of minimal label visiting the arcs of C. If we repeat this procedure, we will finish with a closed path of minimal label and a modified subgraph $\bar{T}$, such that at each vertex the last arc visited by the path is the arc in $\bar{T}$.

We only need to prove $\bar{T}$ is a spanning tree converging to the root, and then by the BEST Theorem, the path is an Eulerian cycle, and by construction it is the one of minimum label.

In order to prove that $\bar{T}$ is a tree (it has no cycles), we prove that the modifications of T breaking a cycle does not produce a new one. Suppose that we modify T at the vertex v adding an arc e in order to break a cycle C, and this addition produces a new cycle C'. Necessarily $C' \bigcap C \neq \emptyset$ so the cycle C' enters to C in a vertex of C different of v. But we are modifying T because all vertices of T except v does not have unvisited arcs not in C, so the arc of C' with head in a vertex of C and tail not in C was already visited by the path and therefore by Lemma 2 all the vertices of $C' \setminus C$ are exhausted by the path, in particular the head of e, which is a contradiction. It remains to prove that every cycle in T has been broken, but this is true by Corollary 7 because all cycles are connected to the subtree of T containing the root vertex, proving the result. □

4 Some Remarks

The previous algorithm does not necessarily produce the minimal label over all Eulerian cycles in all cases, because for some sets of forbidden words does not exist a path starting at the root with the minimal word in the language as label. Precisely, let m and n be respectively the maximal and minimal word of length n in the language, then the previous algorithms produce the minimal de Bruijn sequence if an only if the word mn does not have a factor in $\mathcal{F}$.

However, many of these cases can be solved choosing another vertex as root and using the same tree. The next theorem characterizes from which vertex we can start and to obtain an Eulerian cycle. Note that this theorem is valid for any Eulerian graph.

Theorem 9. *Let T be a spanning tree converging to its root r, let R be the set of vertices having an arc coming from r, and let $\hat{T}$ be the subtree of T defined by $\hat{T} = \bigcup_{u \in R} uTr$. Then a walk W avoiding $(T \cup e)$ starting at v_0 is an Eulerian cycle if and only if $v_0 \in \hat{T}$, where e is the arc from r to the vertex in R in the subtree of v_0 (if it exists).*

Proof. If v_0 is in $\hat{T}$, let e_{v_0} the arc in T with tail at v_0, then $(T \cup e) \setminus e_{v_0}$ is a spanning tree converging to v_0, so a walk avoiding $(T \cup e)$ (and so, avoiding $(T \cup e) \setminus e_{v_0}$) is an Eulerian cycle.

Conversely, suppose W is an Eulerian cycle and $v_0 \notin \hat{T}$. Let v be the vertex in $\hat{T} \bigcap v_0 Tr$ with sons either in $\hat{T}$ or $v_0 Tr$. Applying the Lemma 2, at least all the vertices not in T_{v_0} are exhausted in Wv, in particular all vertices in R, so the

root r is exhausted in Wv, hence all the arcs with head at r are included in Wv and by the same Lemma, v is exhausted in Wv, which is a contradiction. □

To resolve our problem, we modify $\bar{T}$ including the arc with tail at root and head at a floor vertex with maximal label. This will produce a cycle of length $n+1$ because if $m_1 \dots m_n \alpha$ is in the language then $m_i \dots m_n \alpha m_1 m_{i-1}$ is also in the language, so exists in $\bar{T}$ a path of length n between the floor vertex and the root vertex. If one of the vertex in this cycle has a path of length n over $G^{\mathcal{F},n}$ arriving to the vertex of minimal label, by Theorem 9 we can start at this vertex and obtain the Eulerian cycle of minimal label.

Anyway this is not a solution to all cases, figure 3 shows a trivial case where this last strategy does not work. Nevertheless, these are very specific cases.

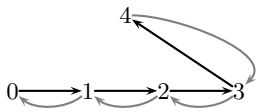

Fig. 3. Example of the graph for $n = 2$, $A = \{0, 1, 2, 3, 4\}$ and $\mathcal{F} = \{00, 02,\ 03, 04, 11,\ 13, 14, 20,\ 22, 24, 30, 31,\ 33, 40, 41,\ 42, 44\}$

Finally, is important to remark that the implementation of the algorithm only takes care of vertices in cycles of T, which can be recognized because the label of these vertices has the structure given by Theorem 6. This condition can be easily checked at each vertex, so the algorithm can be implemented as a linear algorithm in the number of words in the language.

References

1. de Bruijn, N.G.: A combinatorial problem. Nederl. Akad. Wetensch., Proc. **49** (1946) 758–764
2. Stein, S.K.: The mathematician as an explorer. Sci. Amer. **204** (1961) 148–158
3. Bermond, J.C., Dawes, R.W., Ergincan, F.Ö.: De Bruijn and Kautz bus networks. Networks **30** (1997) 205–218
4. Chung, F., Diaconis, P., Graham, R.: Universal cycles for combinatorial structures. Discrete Math. **110** (1992) 43–59
5. Fredricksen, H.: A survey of full length nonlinear shift register cycle algorithms. SIAM Rev. **24** (1982) 195–221
6. Fredricksen, H., Maiorana, J.: Necklaces of beads in k colors and k-ary de Bruijn sequences. Discrete Math. **23** (1978) 207–210
7. Ruskey, F., Savage, C., Wang, T.M.: Generating necklaces. J. Algorithms **13** (1992) 414–430
8. Ruskey, F., Sawada, J.: Generating necklaces and strings with forbidden substrings. Lect. Notes Comput. Sci. **1858** (2000) 330–339

9. Moreno, E.: Lyndon words and de Bruijn sequences in a subshift of finite type. In Harju, T., Karhumäki, J., eds.: Proceedings of WORDS'03. Number 27 in TUCS General Publications, Turku, Finland, Turku Centre for Computer Science (2003) 400–410
10. Lind, D., Marcus, B.: Symbolic Dynamics and Codings. Cambridge University Press (1995)
11. Tutte, W.T.: Graph theory. Volume 21 of Encyclopedia of Mathematics and its Applications. Addison-Wesley Publishing Company Advanced Book Program, Reading, MA (1984)