UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
ESCUELA DE POSTGRADO

TESIS PARA OPTAR AL GRADO DE
DOCTOR EN CIENCIAS DE LA INGENIERÍA
MENCIÓN MODELACIÓN MATEMÁTICA
EN COTUTELA CON LA UNIVERSIDAD DE MARNE-LA-VALLÉE

SECUENCIAS Y GRAFOS DE DE BRUIJN EN LENGUAJES CON RESTRICCIONES

EDUARDO ENRIQUE MORENO ARAYA

2005

UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
ESCUELA DE POSTGRADO

Secuencias y grafos de De Bruijn en lenguajes con restricciones

(De Bruijn graphs and sequences in languages with restrictions)

EDUARDO ENRIQUE MORENO ARAYA

MIEMBROS DE LA COMISIÓN EVALUADORA

Sr. Martín Matamala	Profesor Guía	Universidad de Chile
Sr. Dominique Perrin	Profesor Co-Guía	Université de Marne-la-Vallée
Sr. Julien Cassaigne	Profesor Integrante	Institute de Mathematique de Luminy
Sr. Marcos Kiwi	Profesor Integrante	Universidad de Chile
Sr. Alejandro Maass	Profesor Integrante	Universidad de Chile
Sr. Rafael Correa	Profesor Participante	Universidad de Chile
Sr. Frank Ruskey	Referee	University of Victoria
Sr. Harold Fredricksen	Referee	Naval Postgraduate School

TESIS PARA OPTAR AL GRADO DE
DOCTOR EN CIENCIAS DE LA INGENIERÍA
MENCIÓN MODELACIÓN MATEMÁTICA
EN COTUTELA CON LA UNIVERSIDAD DE MARNE-LA-VALLÉE

SANTIAGO DE CHILE
MAYO 2005

AGRADECIMIENTOS

A Martín Matamala. Han sido 7 años juntos de colaboración y apoyo, los cuales se ven reflejados simbólicamente en esta tesis.

A Alejandro Maass y Dominique Perrin, por su preocupación por mi futuro y por ofrecerme la oportunidad de estudiar en Marne-La-Vallée.

A Marcos Kiwi, por su constante apoyo a lo largo de todos mis estudios y por ser un ejemplo de científico a seguir.

A mis familiares, por forjarme y transformarme en lo que soy.

A mi gran amiga Javiera, y tambien a Pedro y Vicente, por estar junto a mí todos estos años, especialmente en los momentos difíciles.

A Anneli y Violeta, por darme los años más felices de mi vida.

Y no puedo dejar de agradecer a los programas de becas de los Gobiernos de Chile y Francia, por permitirme financiar mis estudios, a los miembros del Departamento de Ingeniería Matemática de la Universidad de Chile y del Institut Gaspard Monge, a Dios, y a todas aquellas personas que anónimamente me han permitido realizar este doctorado.

Índice general

Índice de figuras

RESUMEN DEL INFORME FINAL
PARA OPTAR AL TÍTULO DE
DOCTOR EN CIENCIAS DE LA INGENIERÍA
POR: EDUARDO MORENO ARAYA
PROF. GUÍA: SR. MARTÍN MATAMALA
FECHA DE EXAMEN: 30 MAYO 2005

Secuencias y grafos de De Bruijn en lenguajes con restricciones

Una secuencia de De Bruijn de extensión *n* es una palabra cíclica tal que todas las palabras de un largo fijo *n* aparecen exactamente una vez como factor de la secuencia. Fredricksen y Maiorana construyeron un algoritmo para generar la secuencia de De Bruijn lexicográficamente minimal, el cual utiliza las *palabras de Lyndon*.

Esta tesis estudia cómo generalizar el concepto de secuencias de De Bruijn para un lenguaje compuesto de un subconjunto de palabras de largo *n*, particularmente los lenguajes de todas las palabras de largo *n* sin factores en una lista de *factores prohibidos*.

Inicialmente, se estudia el caso de las palabras sin el factor 11. Se da una demostración del algoritmo de Fredricksen y Maiorana que permite extender este resultado al caso de las palabras sin el factor 1^i para cualquier *i*.

Luego se caracteriza para cuáles lenguajes de palabras de largo *n* existe una secuencia de De Bruijn y se estudia algunas propiedades de la dinámica simbólica de estos lenguajes, particularmente de los lenguajes definidos por factores prohibidos. Para este tipo de lenguajes, se presenta un algoritmo que produce una secuencia de De Bruijn usando las palabras de Lyndon del lenguaje. Este resultado usa la noción de un grafo de De Bruijn y reduce el problema a encontrar un ciclo Euleriano en este grafo.

Por último, se estudia el problema de construir la secuencia de De Bruijn lexicográficamente minimal para un lenguaje con factores prohibidos usando el grafo de De Bruijn. Se estudia dos algoritmos, un algoritmo glotón, simple y eficiente, que funciona para algunos conjuntos de lenguajes, y otro algoritmo más complejo que resuelve este problema para cualquier grafo Euleriano etiquetado.

Résumé

Soit un langage composé par tous les mots d'une longueur donnée *n*. Un cycle de de Bruijn d'ordre *n* est un mot cyclique tel que tous les mots du langage apparaissent exactement une fois comme facteurs de ce cycle. Un algorithme pour construire le cycle de de Bruijn lexicographiquement minimal est dû à Fredricksen et Maiorana, il utilise les mots de *Lyndon* du langage.

Cette thèse étudie comment généraliser le concept de cycles de de Bruijn pour un langage composé par un sous-ensemble de mots de longueur *n*, en particulier les langages de tous les mots de longueur *n* sans facteurs dans une liste de *facteurs interdits*.

Premièrement, nous étudions le cas des mots sans le facteur 11. Nous fournissons de nouvelles preuves de l'algorithme de Fredricksen et Maiorana qui nous permettent de prolonger ce résultat au cas des mots sans le facteur 1^i pour n'importe quel *i*.

Nous caractérisons pour quelles langages de mots de longueur *n* existe un cycle de de Bruijn, et nous étudions également quelques propriétés de la dynamique symbolique de ces langages, en particulier des langages définis par des facteurs interdits. Pour ces genres de langages, nous présentons un algorithme pour produire un cycle de de Bruijn, en utilisant les mots de Lyndon du langage. Ces résultat utilise la notion du graphe de de Bruijn et réduit le problème à construire un cycle eulérien dans ce graphe.

Nous étudions le problème de la construction du cycle minimal dans un langage avec des facteurs interdits en employant le graphe de de Bruijn. Nous étudions deux algorithmes, un algorithme glouton simple et efficace qui fonctionne avec quelques familles de langages, et un algorithme plus complexe qui résout ce problème pour n'importe quel graphe eulérien.

Abstract

Let be a language composed by all words of a given length *n*. A de Bruijn sequence of span *n* is a cyclic string such that all words in the language appears exactly once as factors of this sequence. One of the algorithms to construct the lexicographically minimal de Bruijn sequence is due to Fredricksen and Maiorana and it use the *Lyndon words* in the language.

This thesis studies how to generalize the concept of de Bruijn sequence for a language composed by a subset of words of length *n*, particularly the languages of all words of length *n* without factors in a list of *forbidden factors*.

Firstly, we study the case of words without the factor 11. We give a new proof of the algorithm of Fredricksen and Maiorana which allows us to extend this result to the case of words without the factor 1^i for any i.

We characterize for which languages of words of length *n* exists a de Bruijn sequence, and we also study some symbolic dynamical properties of these languages, particularly of the languages defined by forbidden factors. For these kinds of languages, we present an algorithm to produce a de Bruijn sequence, using the Lyndon words of the language. These results use the notion of de Bruijn graph and reduce the problem to construct an Eulerian cycle in this graph.

We study the problem of construct the lexicographically minimal de Bruijn sequence in a language with forbidden factors using the de Bruijn graph. We study two algorithms, a simple and efficient greedy algorithm which works with some sets of languages, and a more complex algorithm which solves this problem for any Eulerian labelled graph.

INTRODUCTION

Combinatorics on words has grown enormously this last decade. It has now its own section in the lastest classification of Mathematical Reviews (68R15), under the chapter of discrete mathematics related to computer science. Main results in the area have been compiled in the Lothaire's series of books (Lothaire, 1997; Lothaire, 2002; Lothaire, pear) and also since 10 years ago there exists a bi-annual international conference (WORDS) devoted entirely to this subject. To read more about the story and open problems in this subject, see (Berstel and Karhumäki, 2003).

Words appears in almost every area of computer sciences, specially in automata theory, computational complexity and algorithms. One of this area is devoted to the efficient generation of words of a given length. In spite of its specificity, this subject is the title of a whole section of the last volume of Knuth's book "The art of computer programming" (Knuth, pear). Different ways to generate these words have been used. Among the most studied are *Gray codes* (Gray, 1958) and *de Bruijn sequences*. A survey about Gray codes can be found in (Savage, 1997).

In this thesis we focus on the generation of words by using *de Bruijn sequences*. Its first known 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 known as "shift register sequences" and were originally defined by N. G. De Bruijn for the binary alphabet (de Bruijn, 1946). These sequences have many different applications, such as memory wheels in computers and other

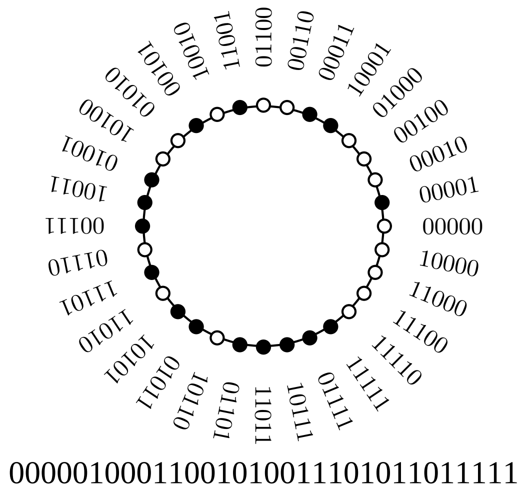


Figure 1.1: De Bruijn sequence of span 5 over the binary alphabet

technological devices, network models, DNA algorithms, pseudo-random number generation, modern public-key cryptographic schemes, to mention a few (Golomb, 1967; Stein, 1961; Bermond et al., 1997a; Chung et al., 1992). The literature about the generation of de Bruijn sequence is extensive (Tuliani, 2001; Hauge and Mykkeltveit, 1996; Mitchell et al., 1996; Ralston, 1981; Etzion, 1986). A good survey in this topic appears in (Fredricksen, 1982).

One of the most interesting and efficient algorithm to construct a de Bruijn sequence is known as the *Fredricksen and Maiorana's Theorem* (sic.) (Fredricksen and Maiorana, 1978). It consists simply in to concatenate in lexicographical order the *Lyndon words* of length dividing *n*. The de Bruijn sequence obtained by this algorithm has another interesting property: it is the lexicographically minimal de Bruijn sequence. This algorithm was proved to be efficient in (Ruskey et al., 1992). An slightly different version of this algorithm was obtained by Duval (Duval, 1983) and proved efficient in (Berstel and Pocchiola, 1994).

In this thesis we study the generation of all words in a set $\mathcal{D}$ of words of length n by using cyclic words. We start in Chapter 2 with $\mathcal{D}$ being the words in $\{0,1\}^n$ not containing the factor 00. In this case we prove that a de Bruijn sequence for $\mathcal{D}$ exists, that is, a cyclic sequence such that its set of factors of length n is the set of words without 00 as factor. Among the exponentially many de Bruijn sequences for $\mathcal{D}$ we also found the lexicographically minimal one. Moreover, in this case the minimal sequence can be constructed efficiently by concatenating Lyndon words of length dividing n without 00 as factor.

The main combinatorial tool used in the study of de Bruijn sequences are *de Bruijn graphs*. Besides its use in the context of de Bruijn sequences, they are also used as models for transportation networks, DNA Algorithms and computer networks to mention a few. The main literature about properties of de Bruijn graphs can be founded in (Bermond et al., 1997b; Lu et al., 2000; Bryant and Fredricksen, 1991). References on generalizations are in (Bermond et al., 1997a; Fredricksen, 1992; Pendavingh et al., 2001; Du and Hwang, 1988). In Chapter 3 we study the existence of de Bruijn sequences for an arbitrary subset $\mathcal{D}$ of words of length n over an alphabet A. In this case, de Bruijn sequence for $\mathcal{D}$ is a cyclic sequence such that its set of factors of length n is exactly $\mathcal{D}$. We first extend the notion of de Bruijn graphs to arbitrary sets $\mathcal{D} \subseteq A^n$. This generalization retains the main properties of de Bruijn graphs, in particular, the existence of de Bruijn sequences for $\mathcal{D}$ is related to the existence of Eulerian trails in the de Bruijn graph for $\mathcal{D}$. Using this representation of de Bruijn sequences, we obtain a characterization of subsets $\mathcal{D}$ admitting a de Bruijn sequence for $\mathcal{D}$ together with an algorithm to construct one of them.

Secondly, by using techniques of symbolic dynamics, we obtain bound for the cardinality of de Bruijn sequences. It is done by enumerating the Eulerian trails over the corresponding de Bruijn graph.

The Fredricksen and Maiorana's algorithm can be seen as a greedy algorithm to find the minimal label among the labels of all Eulerian trails over de Bruijn graphs of span n. In Chapter 4 we start by considering this strategy in de Bruijn graphs for arbitrary sets $\mathcal{D}$ and we obtain a characterization of those sets $\mathcal{D}$ where this strategy produce a minimal de Bruijn sequence. Unfortunately this efficient strategy does not always work.

In the last section of Chapter 4, we consider labelled Eulerian digraphs and the labels of its Eulerian trails. These concepts generalize de Bruijn graphs and de Bruijn sequences, respectively. We present an algorithm to find for a labelled Eulerian digraph the minimal label among all labels of its Eulerian trails.

In spite of the fact that the number of Eulerian trails is exponential with respect to the number of arcs, our algorithm finds the Eulerian trail of minimal label linearly in terms of the number of arcs.

Publications and Structure of the Thesis

The results of each chapter are published or submitted to internatinal journals and conferences. The results of Chapter 2 appears in ADVANCES IN APPLIED MATHEMATICS, published by Elsevier Science. An article with the results of Chapter 3 is accepted in INFORMATION PROCESSING LETTERS (published by Elsevier Science) and a preliminary version of this article has been presented in 4TH INTERNATIONAL CONFERENCE ON COMBINATORICS ON WORDS and published by TUCS General Publications. The results of Chapter 4 appears in two articles, the first one has been presented at the conference WG 2004, GRAPH-THEORETIC CONCEPTS IN COMPUTER SCIENCE and published by SPRINGER LECTURE NOTES IN COMPUTER SCIENCE; the second article has been published as a technical report by CENTER OF MATHEMATICAL MODELING, SANTIAGO, CHILE and is actually submitted to a conference.

Due to copyright restrictions, each chapter includes the full text of each publication, preceded by an introductory section. Exceptionally, in Chapter 3 we append at the end of the chapter some unpublished results presenting some properties about the cardinalities of the combinatorial objects discussed in the chapter. We finish with conclusions and the bibliography of the thesis.

On Fredricksen and Maiorana's Theorem

2.1 Introductory notes

There are many algorithms to construct a De Bruijn sequence *n*. The first known was given by Martin in (Martin, 1934), but the most simple and interesting of this algorithms is the theorem of Fredricksen and Maiorana (Fredricksen and Maiorana, 1978). This theorem proves that the concatenation of the *Lyndon words* of length dividing *n* in lexicographical order, is a de Bruijn sequence of span *n*. Moreover, this sequence is minimal lexicographically between all de Bruijn sequences of span *n*. This algorithm is efficient because Lyndon words can be generated efficiently (Ruskey et al., 1992).

In our interest to generalize the concept of de Bruijn sequence to a particular set of words, we found a particularity of this algorithm. For a given *n*, if we concatenate in lexicographical order the subset of Lyndon words not containing two consecutive 0, we obtain a sequence such that the set of factors of length *n* is the set of cyclic words in A^n not containing two consecutive 0. However, for the symmetrical case (forbidding two consecutive 1) we do not obtain this property.

This remark turned our attention to this theorem. The original proof of this theorem is a greedy algorithm obtaining a de Bruijn sequence. In Section 2.2 we present an alternative proof of the theorem of Fredricksen and Maiorana. In this alternative proof we show how to find the exact position of each factor of length *n* in this de Bruijn sequence. This improvement in the proof also allow us to study the factors of a section of the de Bruijn sequence, which explains the existence of a de Bruijn sequence when we forbid 00 or 11.

For more information about generating de Bruijn sequences and words, see (Sawada and Ruskey, 2003; Cattell et al., 2000; Ruskey and Sawada, 1999; Sawada, 2001; Sawada, 2003; Shiloach, 1981; Tuliani, 2001; Hauge and Mykkeltveit, 1996; Mitchell et al., 1996; Stein, 1961; de Bruijn and Erdös, 1948; Chung et al., 1992; Berstel and Pocchiola, 1994; Fredricksen and Kessler, 1977).

2.2 Full paper:

On the Theorem of Fredricksen and Maiorana about de Bruijn Sequences

In: Advances in Applied Mathematics

Volume 33, Number 2, Pages 413–415

doi:10.1016/j.aam.2003.10.002

Abstract

This work gives an alternative proof for the theorem of Fredricksen and Maiorana (Discrete Mathematics, 23 (1978), 207-210) about constructing a de Bruijn sequence by concatenation of the Lyndon words in lexicographic order. This proof gives the exact position of all the words in the sequence, and allows us to extend this result to the concatenation of any number of the last Lyndon words in increasing order.

2.2.1 Introduction

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 for a *suffix* is symmetrical.

The set A^* of all the words on the alphabet A is linearly ordered by the lexicographic order induced by $<$. By definition, $x < y$ if either 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 lexicographic order is the following: if $x < y$ and if x is not a prefix of y, then $xu < yv$ for all words u, v.

Two words x, y are *conjugated* if there exist words u, v in A^* 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 u^n for $u \in A^*$ and $n \geq 2$. A *Lyndon word* is a word that is both primitive and minimal.

A de Bruijn sequence of span n is a string B^n of length $|A|^n$ such that all the words of length n are present as substrings of B^n exactly once. A very important question about de Bruijn sequence is how to generate them efficiently (see (Fredricksen, 1982) for a survey in this subject). One of the most efficient and elegant solutions to this problem is given in (Fredricksen and Maiorana, 1978), and it is achieved by the concatenation in lexicographic order of the Lyndon words whose lengths divide n.

In this work, we give an alternative proof of this theorem giving the exact position of all words in the sequence, and presenting some conditions for the existence of a de Bruijn sequence by concatenation of Lyndon words. These conditions allow us to conclude on the construction of a de Bruijn sequence by concatenating any number of the last Lyndon words in the lexicographic order.

2.2.2 The theorem of Fredricksen and Maiorana

Theorem 2.2.1. *For a given n, the lexicographic concatenation of Lyndon words of length dividing n generates a de Bruijn sequence of span n.*

Proof. Let a and z be the minimal and the maximal letters in the alphabet A, let σ be the usual shift operator and let B^n be the de Bruijn sequence of span n.

We will prove that for any minimal word w of length n, all its conjugated words $\sigma^i(w)$, $i = 0 \dots n-1$, are substrings 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 the last $n - j$ conjugated words are substrings of B^n. Note that these words have the form $z^i w_1 \dots w_j z^{n-j-i}$, for $i = 1 \dots n-j$.

Let v be the minimal Lyndon word with prefix $w_1 \dots w_j z^{n-j-i}$. Then, the previous minimal word in the lexicographic order has the form $u = u_1 \dots u_{n-i} z^i$ with $u_1 \dots u_{n-i} < w_1 \dots w_j z^{n-j-i}$. Hence, the Lyndon word before v has suffix z^i, and then $z^i w_1 \dots w_j z^{n-j-i}$ is a substring of B^n.

Now we will prove this for the first $j-1$ rotations. If w is not a Lyndon word (is not primitive), let $\overline{w}$ be the primitive root of w and let l be its length. Note that $\overline{w}$ has the form $\overline{w}_1 \dots \overline{w}_{j'} z^{l-j'}$ with $l - j' = n - j$. If $\overline{w} \neq z$ then the next Lyndon word in lexicographic order x has the form $x = \overline{w}^{\frac{n}{l}-1} w_1 \dots w_{j'-1}(w_{j'}+1) b_{j'+1} \dots b_l$, so $\sigma^i(w)$ is a substring of $\overline{w}x$ for $i = 0 \dots j-1$. The case $\overline{w} = z$ is trivial.

If w is primitive, let x be the next minimal word in lexicographic order (not necessarily primitive). Therefore x has the form $x_1 \dots x_{j-1}(x_j+1) b_{j+1} \dots b_n$ and in this case $\sigma^i(w)$ is a substring of wx for $i = 0 \dots j-1$. If x is primitive, then wx is a substring of B^n and we are done, otherwise, by the previous argument, x is a prefix of $\overline{x}y$ where y is the next Lyndon word in lexicographic order, hence wx is a substring of $w\overline{x}y$ and therefore it is a substring of B^n. □

Given a set $\{L_i\}_{i \in I}$ of Lyndon words whose lengths divide n, we can naturally associate a language to this set composed by words of lengths n corresponding to all conjugated words of L_i, if $|L_i| = n$, and all conjugated words of $(L_i)^r$, if $|L_i| = n/r$. A de Bruijn sequence generating this language will be called a “partial”-de Bruijn sequence.

Those languages are interesting because given a set $\mathcal{F}$ of forbidden blocks, the language of circular words not having a substring in $\mathcal{F}$ is exactly the language associated to Lyndon circular words not having a substring in $\mathcal{F}$. This kind of language is know as *subshift of finite type* and is the fundamental concept of the symbolic dynamic.

Corollary 2.2.2. *Let $L_1,\ldots,L_m$ be the Lyndon words of A^n of length dividing n ordered in lexicographic order, then for any $s<m$, $L_sL_{s+1}\cdots L_m$ is a "partial"-de Bruijn sequence.*

Proof. We will prove that all rotations of a minimal word w are substrings of the sequence.

If w is not a Lyndon word (i.e. if it is a power of a Lyndon word L_k with $|L_k|<n$) then by the previous proof we know that all rotations of w are contained in $L_{k-1}L_kL_{k+1}$. Hence, we only have to check the case $i=s$, but in this case, for $m>2$ we have that z^n is a suffix of the sequence, and then we have enough z letters to obtain all the rotations of w beginning with z.

Let $w=w_1\ldots w_jz^{n-j}$ be a Lyndon word L_k with $w_j<z$. Again by the previous proof, we know that the first $j-1$ rotations of w are contained in $L_kL_{k+1}\ldots L_m$ and so we only need to check the last $n-j$ conjugated words, which have the form $z^iw_1\ldots w_jz^{n-j-i}$ for $i=1\ldots n-j$.

By the previous proof, if the minimal Lyndon word having prefix $w_1\ldots w_j$ is included in the sequence, then we know that all the rotations of w are substrings of the sequence, on the contrary we would have $w_1\ldots w_j<L_s\leq w_1\ldots w_jz^{n-j}$, which means that the first Lyndon word in the sequences has the form $L_s=w_1\ldots w_jb_{j+1}\ldots b_n$ and therefore $z^{n-j-i}w_1\ldots w_j$ is a substring of the sequence.

It remains to check the rotations of the form $z^iw_1\ldots w_jz^{n-j-i}$ for $i=1\ldots n-j-1$. If the minimal Lyndon word having prefix $w_1\ldots w_jz$ is included in the sequence then we are done. If not, $w_1\ldots w_jz<L_s\leq w1\ldots w_jz^{n-j}$, in which case $L_s=w_1\ldots w_jzb_{j+2}\ldots b_n$.

Therefore we conclude that $z^{n-j-1}w_1 \dots w_j z$ is a substring of the sequence.

We can repeat this argument until have a minimal Lyndon word with prefix $w_1 \dots w_j z^t$, in which case all the remaining rotation $z^i w_1 \dots w_j z^{n-j-i}$ for $i = 1 \dots n - j - t$ will be substrings of the sequence. We know that such a t exists because $L_S \leq w_1 \dots w_j z^{n-j}$. □

GENERALIZATION OF DE BRUIJN SEQUENCES AND DE BRUIJN GRAPHS

3.1 Introductory notes

In the Chapter 2 we construct the de Bruijn sequence generating all words of a fixed length without the factor 00. This fact motivates us to generalize the definition of de Bruijn sequence in order to generate a particular set of words $\mathcal{D}$ of a fixed length.

As in the example of Chapter 2, we are interested especially in sets $\mathcal{D}$ of words defined as the set of words without a (one or more) *Forbidden Factor*. These sets of words are known as the languages of *subshifts of finite type* and is the basic object in the area of symbolic dynamics.

In this chapter we present a generalization of de Bruijn sequence in order to study these cases. It is not difficult to see that only some sets $\mathcal{D}$ have a de Bruijn sequence. Hence we characterize the sets $\mathcal{D}$ with a de Bruijn sequence.

In the case of subshifts of finite type, we characterize the sets of forbidden words such that the language obtained has a de Bruijn sequence, and we give an algorithm to produce a de Bruijn sequence producing the periodical words in the language of a subshift of finite type.

All this results use a graph-theoretic interpretation of de Bruijn sequence, based in the *de Bruijn graph*, which appeared first implicitly in (Flye Sainte-Marie, 1894) and explicitly in (de Bruijn, 1946; Good, 1946) independently. Among its many applications, the principal one is as a model for transportation networks. That is because de Bruijn graph has an interesting property: the shortest route from one vertex to another is completely defined by the labels of the vertices. This property is used all along the paper associated to this chapter and is one of the main ingredient in order to prove our results.

For more information about the properties of de Bruijn graphs, see (Bermond et al., 1997b; Bermond et al., 1997a; Lu et al., 2000; Bryant and Fredricksen, 1991; Fredricksen, 1992; Mykkeltveit, 1972)

3.2 Full paper:

De Bruijn Sequences and De Bruijn Graphs for a General Language

Accepted at: Information Processing Letters

Abstract

A de Bruijn sequence over a finite alphabet of span n is a cyclic string such that all words of length n appear exactly once as factors of this sequence. We extend this definition to a subset of words of length n, characterizing for which subsets exists a de Bruijn sequence. We also study some symbolic dynamical properties of these subsets extending the definition to a language defined by forbidden factors. For these kinds of languages we present an algorithm to produce a de Bruijn sequence. In this work we use graph-theoretic and combinatorial concepts to prove these results.

3.3 Introduction

Given a set $\mathcal{D}$ of words of length n, a de Bruijn sequence of span n is a periodic sequence such that every word in $\mathcal{D}$ (and no other n-tuple) occurs exactly once. Its first known 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 known as “shift register sequences” and were originally studied by N. G. De Bruijn for $\mathcal{D} = \{0,1\}^n$ (de Bruijn, 1946). These sequences have many different applications, such as memory wheels in computers and other technological devices, network models, DNA algorithms, pseudo-random number generation, modern public-key cryptographic schemes, to mention a few (see (Stein, 1961; Bermond et al., 1997a; Chung et al., 1992)). Typically, de Bruijn sequences have been studied over an arbitrary alphabet A considering the set of all the n-tuples, that is, A^n. There is an exponential number of de

Bruijn sequences in this case, but only a few can be generated efficiently.

In this work we generalize the definition of de Bruijn sequence for any set $\mathcal{D}$, characterizing those sets $\mathcal{D}$ for which a de Bruijn sequence exists. In section 3.5 we study some symbolic dynamical properties of these sets, extending our results to languages defined by forbidding some factors. Finally in section 3.6 we present an algorithm to produce a de Bruijn sequence for these kinds of languages.

3.4 Definitions and generalizations

Let A be a finite set. A *word* w on the alphabet A is a finite sequence of elements of A. For a word w, its length is denoted by $|w|$.

A word p is said to be a *factor* of a word w if there exist words $u, v \in A^*$ such that $w = upv$. If u is the empty word (denoted by ε), then p is called a *prefix* of w, and if v is empty then is called a *suffix* of w.

Let $\mathcal{D}$ be a set of words of length $n+1$. We call this set a *dictionary*. A *de Bruijn sequence of span* $n+1$ *for* $\mathcal{D}$ is a cyclic word $B^{\mathcal{D}}$ of length $|\mathcal{D}|$ such that all the words in $\mathcal{D}$ are factors of $B^{\mathcal{D}}$. In other words,

$$\{(B^{\mathcal{D}})_i \dots (B^{\mathcal{D}})_{i+n \mod |\mathcal{D}|} | i = 0, \dots, |\mathcal{D}| - 1\} = \mathcal{D}$$

De Bruijn sequences are closely related to de Bruijn graphs. The *de Bruijn* *graph of span n for $\mathcal{D}$*, denoted by $G^{\mathcal{D}}$, is the directed graph with vertex set

$$V(G^{\mathcal{D}}) = \{u \in A^n | u \text{ is a prefix or a suffix of a word in } \mathcal{D}\}$$

and arc set

$$E(G^{\mathcal{D}}) = \{(\alpha v, v\beta) | \alpha, \beta \in A, \alpha v \beta \in \mathcal{D}\}$$

This graph was first defined implicitly in 1894 by Flye (Flye Sainte-Marie, 1894) and it was explicitly detailed in 1946 by de Bruijn (de Bruijn, 1946) and Good (Good, 1946) independently. In both cases the dictionary studied was $\mathcal{D} = A^{n+1}$. The first use of this graph for a subset of A^{n+1} was given in (Rauzy, 1983).

From this definition, we can do a bijection between the arcs of $G^{\mathcal{D}}$ and the words in $\mathcal{D}$: to an arc going from αv to $v\beta$ we associate the word $\alpha v \beta$. Using this bijection we can interpret the graph $G^{\mathcal{D}}$ as the union of non-trivial components of the original de Bruijn graph for A^{n+1} after removing the arcs corresponding to words not in $\mathcal{D}$ (see Figure 3.1).

We label the graph $G^{\mathcal{D}}$ using the following function l: if $e = (\alpha v, v\beta)$ then $l(e) = \beta$. This labeling has an interesting property:

Remark 3.4.1. Let $P = e_0 \ldots e_m$ be a walk over $G^{\mathcal{D}}$ of length $m \geq n$. Then P finishes in a vertex u if and only if u is a suffix of $l(P) = l(e_0) \ldots l(e_m)$.

This property is essential to understand de Bruijn graphs and will be used in all the proofs in this work. Therefore we mention a few important consequences of this property:

Corollary 3.4.2. *All the walks of length $n+1$ finishing at vertex u have label αu for some*

$\alpha \in A$.

Corollary 3.4.3. *If u and v are vertices of a cycle C, then u and v are factors of the infinite word $l(C)^\infty$.*

These consequences and the bijection between arcs and words in $\mathcal{D}$ explain the relation between de Bruijn graphs and de Bruijn sequences:

Lemma 3.4.4. *There exists a de Bruijn sequence $B^{\mathcal{D}}$ if and only if $G^{\mathcal{D}}$ is an Eulerian graph. Moreover, the labels of Eulerian cycles over $G^{\mathcal{D}}$ are the de Bruijn sequences for $\mathcal{D}$.*

Proof. Let C be an Eulerian cycle of $G^{\mathcal{D}}$. As we explained before, any word $w \in \mathcal{D}$ has a corresponding arc e in $G^{\mathcal{D}}$. By Remark 3.4.1 any sub-walk of length $n+1$ of C finishing with the arc e has label w, therefore any word in $\mathcal{D}$ is a factor of $l(C)$. As the length of C is the number of words in $\mathcal{D}$ we conclude that $l(C)$ is a de Bruijn sequence for $\mathcal{D}$.

Conversely, let B be a de Bruijn sequence for $\mathcal{D}$. Any factor of length $n+1$ is a word of $\mathcal{D}$ so there is a corresponding arc in $G^{\mathcal{D}}$. Moreover, two consecutive factors αv and $v\beta$ have two corresponding arcs such that the head of the first is the tail of the second one. Therefore B has a corresponding closed walk over $G^{\mathcal{D}}$ with label B. Since every factor is different, every arc in the walk is different, and since every word of $\mathcal{D}$ is a factor of B, every arc of $G^{\mathcal{D}}$ is in the walk. We conclude that the closed walk over $G^{\mathcal{D}}$ is an Eulerian cycle of label B □

By previous lemma, given a dictionary $\mathcal{D}$, the existence of a de Bruijn sequence of span $n+1$ is characterized by the existence of an Eulerian cycle over $G^{\mathcal{D}}$. A graph has an Eulerian cycle if and only if it is strongly connected and at each vertex the in-degree and the out-degree are equal. Therefore we can write these conditions as restrictions over $\mathcal{D}$, characterizing the dictionaries with a de Bruijn sequence.

Corollary 3.4.5. *A dictionary $\mathcal{D} \subseteq A^{n+1}$ has a de Bruijn sequence of span $n+1$ if and only if*

1. *For any $u, v \in \mathcal{D}$ there exists a word $w \in A^*$ such that u is a prefix of w, v is a suffix of w and any factor of length $n+1$ of w is in $\mathcal{D}$.*
2. *For any word $x \in A^n$ there exists a bijection between words in $\mathcal{D}$ having x as a suffix, and words in $\mathcal{D}$ having x as a prefix.*

Proof. By the bijection between arcs and words in $\mathcal{D}$, the first condition assures the existence of a walk (of label w) between any two arcs. Hence $G^{\mathcal{D}}$ is strongly connected. For any word x, a word in $\mathcal{D}$ having x as suffix (prefix) has a corresponding arc terminating (starting) at x. Therefore, the second condition assures that the in-degree and the out-degree at any vertex are equal □

3.5 Symbolic dynamics

Symbolic dynamics gives a natural framework to study the sets $\mathcal{D}$ with a de Bruijn sequence.

A first class of dictionaries with a de Bruijn sequence is given by the set of factors of length $n+1$ in a bi-infinite sequence $\mathbf{u}$ over an alphabet A. We denote this set by $\mathcal{L}_{n+1}(\mathbf{u})$.

A factor v of length n in $\mathbf{u}$ is *right extensible* (resp. left extensible) if $v\alpha$ (resp. αv) is in $\mathcal{L}_{n+1}(\mathbf{u})$ for some $\alpha \in A$. These concepts have an important relation with the complexity of the sequence (see (Cassaigne, 1997)).

For any sequence $\mathbf{u}$, is easy to see that the dictionary $\mathcal{D} = \mathcal{L}_{n+1}(\mathbf{u})$ satisfies the first condition of Corollary 3.4.5. Also, the second condition is satisfied if and only if the numbers of left and right extensions of any factor of length n are equal. Therefore, we obtain the next theorem.

Theorem 3.5.1. *Let $\mathbf{u}$ be a bi-infinite sequence. For any n, the dictionary $\mathcal{D} = \mathcal{L}_{n+1}(\mathbf{u})$ has a de Bruijn sequence if and only if any factor of length n has equal number of left and right extensions.*

Another class of dictionaries with a de Bruijn sequence is given by the language of subshifts. Given an alphabet A, a full shift $A^{\mathbb{Z}}$ is the set of all bi-infinite sequences of symbols from A. Let $\mathcal{F}$ be a collection of (finite) words, we call these words "forbidden words". A shift $X = X_{\mathcal{F}}$ is the subset of sequences of $A^{\mathbb{Z}}$ which do not contain any factor from $\mathcal{F}$. If $\mathcal{F}$ is finite, we say that X is a subshift of finite type.

Let $\mathcal{L}_n(X)$ be the set of factors of sequences in X of length n. The language of a shift X is the set $\mathcal{L}$ of the factors of any finite length of sequences in X.

$$\mathcal{L}(X) = \bigcup_{n=0}^{\infty} \mathcal{L}_n(X)$$

A shift X is irreducible if for every pair of words $u, v \in \mathcal{L}(X)$, there is a $w \in \mathcal{L}(X)$ such that $uwv \in \mathcal{L}(X)$.

Given a labeled graph G, let X_G be the set of labels of all bi-infinite walks over G. It is known that X_G is a (sofic) shift (Lind and Marcus, 1995), however in the case of de Bruijn graphs, we show that $X_{G^{\mathcal{D}}}$ is a subshift of finite type.

Lemma 3.5.2. *Let $\mathcal{D} \subseteq A^{n+1}$ be a dictionary. Then $X_{G^{\mathcal{D}}}$ is a subshift of finite type. Moreover,*

$$X_{G^{\mathcal{D}}} = X_{\mathcal{F}} \text{ with } \mathcal{F} = A^{n+1} \setminus \mathcal{D}$$

Proof. Since $\mathcal{L}_{n+1}(X_{G^{\mathcal{D}}}) \subseteq \mathcal{D}$ we have that $X_{G^{\mathcal{D}}} \subseteq X_{\mathcal{F}}$. Let $x \in X_{\mathcal{F}}$, any factor of length $n+1$ of x is in $\mathcal{D}$ so each factor has a corresponding arc in $G^{\mathcal{D}}$. Moreover, two consecutive factors αv and $v\beta$ of length $n+1$ have two corresponding arcs in $G^{\mathcal{D}}$ such that v is the head of the first and the tail of the second one. Therefore there exists a walk over $G^{\mathcal{D}}$ with label x, so $X_{\mathcal{F}} \subseteq X_{G^{\mathcal{D}}}$ □

Corollary 3.5.3. *Let $\mathcal{F}$ be a set of forbidden words of length at most $n+1$. Then for $\mathcal{D} = \mathcal{L}_{n+1}(X_{\mathcal{F}})$ we have that $X_{\mathcal{F}} = X_{G^{\mathcal{D}}}$.*

Proof. We can extend $\mathcal{F}$ to a subset $\mathcal{F}' \subseteq A^{n+1}$ such that $X_{\mathcal{F}} = X_{\mathcal{F}'}$. Since $\mathcal{D} = A^{n+1} \setminus \mathcal{F}'$ we conclude □

A vertex v is *stranded* if either no arc starts at v or no arc terminates at v. A subgraph is *essential* if no vertex of the graph is stranded (see Figure 3.1). Obviously a bi-infinite walk does not use stranded vertices, so for any graph G there exists an essential subgraph G' such that $X_G = X_{G'}$. Therefore in the rest of this work we only consider sets $\mathcal{D}$ such that $G^{\mathcal{D}}$ is essential.

Note that if $G^{\mathcal{D}}$ is essential then for any word $w \in \mathcal{D}$ there exists a walk over $G^{\mathcal{D}}$ with label w.

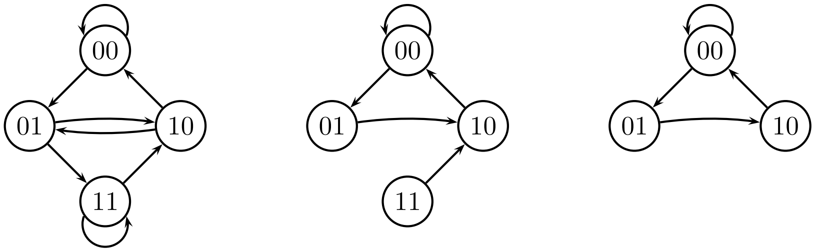


Figure 3.1: Examples in a binary alphabet: De Bruijn sequence of span 2, $G^{\mathcal{D}}$ for $\mathcal{D} = \{000, 001, 010, 100, 110\}$ and its essential subgraph.

In order to obtain sets $\mathcal{D}$ with a de Bruijn sequence, $G^{\mathcal{D}}$ needs to be an Eulerian graph, in particular it needs to be strongly connected. This property has an interpretation in symbolic dynamics:

Lemma 3.5.4. *Let $\mathcal{D}$ be a dictionary. $X_{G^{\mathcal{D}}}$ is irreducible if and only if $G^{\mathcal{D}}$ is strongly connected.*

Proof. Since $G^{\mathcal{D}}$ is essential, the strongly connected components have size at least 2. Let $X_{G^{\mathcal{D}}}$ be an irreducible subshift of finite type. For any two arcs e, f of $G^{\mathcal{D}}$ there are two corresponding words w_e, w_f in $\mathcal{D}$. Since $X_{G^{\mathcal{D}}}$ is irreducible, there exists a word $\hat{w}$ such that $w_e\hat{w}w_f$ is a factor of $X_{G^{\mathcal{D}}}$. In other words, there exists a walk over $G^{\mathcal{D}}$ with label $w_e\hat{w}w_f$. Therefore there exists a walk with label $\hat{w}w_f$ connecting e to f, so $G^{\mathcal{D}}$ is strongly connected.

Conversely, if $X_{G^{\mathcal{D}}}$ is not irreducible, there exist factors w_1, w_2 such that $\forall z \in A^*$, w_1zw_2 is not a factor of $X_{G^{\mathcal{D}}}$. But w_1 is the label of a walk over $G^{\mathcal{D}}$ finishing at a vertex v_1 and w_2 is the label of a walk starting at a vertex v_2, therefore there is no walk over $G^{\mathcal{D}}$ connecting v_1 to v_2, hence $G^{\mathcal{D}}$ is not strongly connected □

Let $X_{\mathcal{F}}$ be an irreducible subshift of finite type. If $\mathcal{D} = \mathcal{L}_{n+1}(X_{\mathcal{F}})$ then the corresponding graph $G^{\mathcal{D}}$ is not necessarily an Eulerian graph.

For example, for $A = \{0,1\}$ and $\mathcal{F} = \{11\}$ a vertex $0w1$ has two in-going arcs (corresponding to words $00w1$ and $10w1$) but only one out-going arc (corresponding to the word $0w10$). Therefore we will study the subset of *periodic* words in $\mathcal{L}_{n+1}(X_{\mathcal{F}})$ because for this set we obtain an Eulerian de Bruijn graph.

Let $w \in A^*$ be a word, we say that w is a *periodic* word of $X_{\mathcal{F}}$ if and only if the bi-infinite sequence w^∞, obtained by infinite concatenations of w, is in $X_{\mathcal{F}}$. The set of periodic words of length n is denoted by $\mathcal{P}_n(X_{\mathcal{F}})$.

Theorem 3.5.5. *Let $\mathcal{F}$ be a set of forbidden words of length at most $n+1$ and $\mathcal{D} = \mathcal{P}_{n+1}(X_{\mathcal{F}})$. If $X_{G^{\mathcal{D}}}$ is irreducible then there exists a de Bruijn sequence for the dictionary $\mathcal{D}$.*

Proof. By Lemma 3.5.4, $G^{\mathcal{D}}$ is strongly connected. Let $u \in A^n$ be a vertex of $G^{\mathcal{D}}$. Any arc leaving u with label α corresponds to a word $u\alpha \in \mathcal{D}$. Since $u\alpha$ is a periodic word, αu is also in $\mathcal{D}$. Therefore there exists an arc going into u corresponding to the word αu, which implies that the in-degree of u is greater or equal to out-degree of u. The same argument proves that the out-degree of u is greater than or equal to the in-degree of u, concluding that $G^{\mathcal{D}}$ is an Eulerian graph □

Note that not all irreducible subshifts of finite type have a de Bruijn sequence for $\mathcal{D} = \mathcal{P}_{n+1}(X_{\mathcal{F}})$. For example, for $A = \{0,1\}$ and $\mathcal{F} = \{010\}$ the subshift of finite type $X_{\mathcal{F}}$ is irreducible but $X_{G^{\mathcal{D}}}$ is not irreducible, because $G^{\mathcal{D}}$ has two strongly connected components.

3.6 Constructing a de Bruijn sequence for subshifts.

Let $X_{\mathcal{F}}$ be a subshift of finite type and $\mathcal{D} = \mathcal{P}_{n+1}(X_{\mathcal{F}})$ such that $X_{G^{\mathcal{D}}}$ is irreducible. In this section we study an efficient generation of a de Bruijn sequence for $\mathcal{D}$.

Even in the unrestricted case (where $\mathcal{F} = \emptyset$) this is an interesting problem (see (Fredricksen, 1982) for a survey on this subject). One of the most elegant and efficient solutions in the unrestricted case is given in (Fredricksen and Maiorana, 1978) and uses *Lyndon words*.

Let $<$ be a linear order over alphabet A. The set A^* of all words on the alphabet A is linearly ordered by the lexicographical order induced by the order $<$ on A. A word w is a Lyndon word if and only if $\forall u, v$ such that $w = uv$, then $w < vu$.

The algorithm of Fredricksen and Maiorana consists of to concatenate in increasing lexicographical order the Lyndon words of length dividing n. This is a linear time algorithm because the Lyndon words can be generated efficiently (see (Ruskey et al., 1992)).

We always can construct the graph $G^{\mathcal{D}}$ and apply one of the known results about constructing an Eulerian cycle to obtain a de Bruijn sequence, however the construction of $G^{\mathcal{D}}$ is not efficient. Therefore in this section we study the structure of $G^{\mathcal{D}}$ in order to obtain an algorithm to construct a de Bruijn sequence only using the words in $\mathcal{D}$.

The set of arcs of an Eulerian graph can be partitioned in cycles. In the particular case of $G^{\mathcal{D}}$ these cycles have a given length.

Theorem 3.6.1. *Let $\mathcal{F}$ be a set of forbidden words of length at most $n+1$ and $\mathcal{D} =$*

$\mathcal{P}_{n+1}(X_{\mathcal{F}})$ such that $G^{\mathcal{D}}$ is the de Bruijn graph of span n for $\mathcal{D}$. Then the cycles of length dividing $n+1$ partition the set of arcs of $G^{\mathcal{D}}$.

Proof. We prove that any arc of the graph is in one and only one cycle of length dividing n.

Let e be an arc from the vertex au to the vertex ub with $a,b \in A$ (then, $l(e) = b$). By construction of the graph, there is a walk of length n from vertex ub to vertex au with label au. Therefore, the union of this walk with the arc e produces a closed walk of length $n+1$ with label aub corresponding to one or more repetitions of a cycle of length dividing $n+1$, proving the existence of one cycle.

Let us suppose now that there are two cycles C and C' of lengths dividing $n+1$ using the arc e. Let f be an arc of C and g an arc of C' with tail at the same vertex u and different heads. Since e is in both cycles, by Corollary 3.4.2 the walks of length n from the head of e to the tail of e using only the arcs of C and C' must have the same label. Therefore the label of $l(f) = l(g)$ but in this case the head of f and the head of g are the same vertex, producing a contradiction. This proves the uniqueness of the cycles □

Corollary 3.6.2. *The set of Lyndon words of length dividing $n+1$ in $\mathcal{L}(X_{\mathcal{F}})$ corresponds to a partition of the set of arcs of $G^{\mathcal{D}}$.*

Proof. Let C be a cycle of length d with d dividing $n+1$ and let us label it w in such a way that $\forall u, v$ such that $w = uv$, we have that either $w = vu$ or $w < vu$. We only have to prove that w is not a repetition of a smaller word u.

Let us assume that $w = u^i$ for an integer $i \geq 2$ and let x and y be two vertices of C at distance $|u|$ over C such that the walk of C from x to y has label u. Since both vertices are in C, x and y are factors of length n of the word $w^{(n+1)/d}$. Since the walk from x to y has label u, u is a suffix of y. Moreover, since $w^{(n+1)/d} = u^{i(n+1)/d}$, uu is a suffix of y then u is also a suffix of x, concluding that $x = y$.

Therefore, every cycle in the partition has a different label which is a Lyndon word of length dividing $n+1$.

It remains to prove that to each Lyndon word, one can associate a cycle. But this can be proved using cardinality considerations. Indeed, a periodic word of length $n+1$ has either least period $n+1$ or least period d with d dividing $n+1$. Therefore,

$$|\mathcal{P}_{n+1}(X_{\mathcal{F}})| = \sum_{d|n+1} |\{\text{words with least period } d\}|$$

Now, a word with least period d is a Lyndon word or one of the $d-1$ rotations of a Lyndon word of length d. Hence,

$$\sum_{d|n+1} |\{\text{words with least period } d\}| = \sum_{d|n+1} d \cdot |\{\text{Lyndon words of length } d\}|$$

Since $|E(G^{\mathcal{D}})| = |\mathcal{P}_{n+1}(X_{\mathcal{F}})|$ we conclude. □

Now we are prepared to construct an algorithm producing a de Bruijn sequence for $\mathcal{D} = \mathcal{P}_{n+1}(X_{\mathcal{F}})$.

Given a partition in cycles of an Eulerian graph, the following strategy produces an Eulerian cycle: we can start from an arc and follow the corresponding cycle in the partition, until we reach an intersection with another cycle in the partition. At this point we follow the other cycle and when we return to the intersection we continue with the original cycle. Using this procedure recursively we construct an Eulerian cycle.

By Corollary 3.6.2, we can reproduce this strategy in terms of the Lyndon words of length diving $n+1$ in $\mathcal{L}(X_{\mathcal{F}})$ obtaining Algorithm 3.6.1 producing a de Bruijn sequence for $\mathcal{D} = \mathcal{P}_{n+1}(X_{\mathcal{F}})$ without constructing the graph $G^{\mathcal{D}}$.

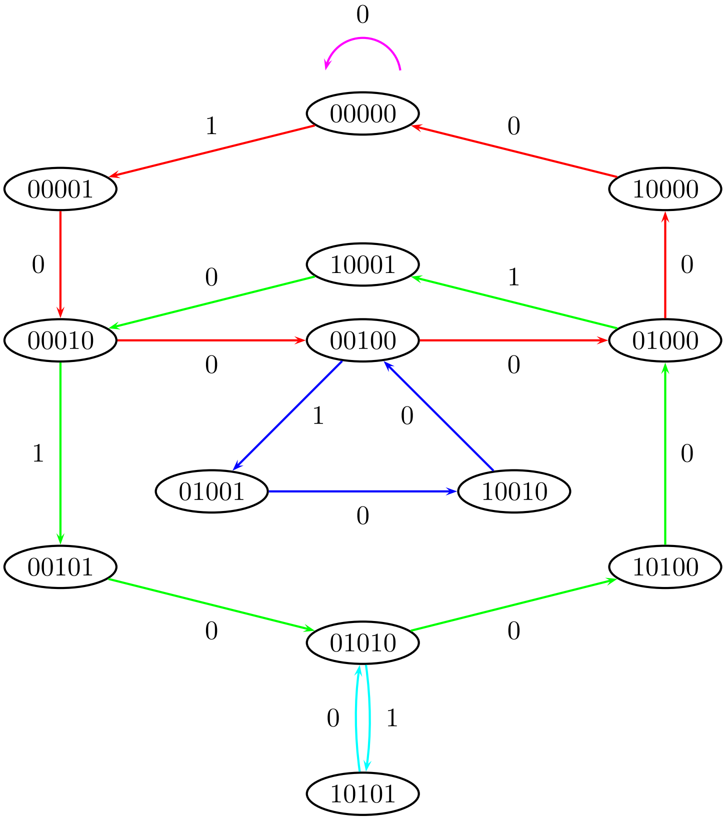
Algorithm 3.6.1: Produce a de Bruijn sequence using the Lyndon words of the language.

```
INPUT: $L = \{L^1, \dots, L^k\}$ Lyndon words in $\mathcal{L}(X_{\mathcal{F}})$ of length dividing $n+1$.
(1)   $Size \leftarrow \sum |L^i|$
(2)   $u \leftarrow L^i$ for any $L^i \in L$ such that $|L^i| = n+1$
(3)   $L \leftarrow L \setminus u$
(4)   $B \leftarrow uu$
(5)   while $L \neq \emptyset$
(6)     for $\alpha = 1$ to $|A| - 1$
(7)       $w \leftarrow B_{j-n-1} \dots B_j \overline{B_{j+1}}^{\alpha}$
(8)       $w' \leftarrow$ LYNDON$(w)$
(9)       if $w' \in L$ then
(10)        $B \leftarrow B_1 \dots B_j w_n w_1 \dots w_{|w'|-1} B_{j+1} \dots$
(11)        $L \leftarrow L \setminus w'$
(12)      end if
(13)    end for
(14)  end for
(15)  $B \leftarrow B_1 \dots B_{Size}$
```

where $\overline{a}^{\alpha} = a + \alpha \mod |A|$ and LYNDON(w) return the Lyndon word z such that $z^{\infty} = w^{\infty}$.

The function LYNDON() in the algorithm can be implemented with an online automata accepting when a suffix of B is a factor of length n of rotations of the words in L, allowing to do this step in a constant time (see (Crochemore et al., 2001)). Hence, steps $(7-12)$ in the algorithm have complexity $O(n)$ and these steps are repeated at most $|A| \cdot |L|$ times. Therefore, the complexity of the algorithm is $O(|A| \cdot |L| \cdot n)$. Since $Size = \sum_L |L^i|$ is the size of the input (and also the size of the output) and $Size$ is at most $n \cdot |L|$, we conclude that our procedure is a linear time algorithm. Note that the input of the algorithm can also be constructed in an efficient way (see (Ruskey and Sawada, 2000)).

Example 3.6.3. Example of the algorithm for the Golden Mean system ($\mathcal{F} = \{11\}$) with length $n = 6$ and Lyndon words $L = \{0, 000001, 000101, 001, 01\}$. In the figure we can see the partition in cycles of the de Bruijn graph corresponding to the Lyndon words in L. The graphical representation of the algorithm starts at vertex 00001.



000001	We start with the first word in *L*.
$\underline{\downarrow 0}\underline{00001}$	In this case 000011 is not in *L*.
$\underline{0 \downarrow 0}\underline{0001}$	here 000101 is in *L*, we include (100010)
0(100010)00001	
$\underline{01 \downarrow 0}001000\underline{001}$	here 001011 is not in *L*
$\underline{010 \downarrow 0}010000\underline{01}$	here 010101 is in *L*, we include (10)
0(10(10)0010)00001	
$\underline{0101 \downarrow 0}00100000\underline{1}$	101011 is not in *L*
$\underline{01010 \downarrow 0}01000001$	010101 was already included
$0\underline{10100 \downarrow 0}1000001$	101001 is not in *L*
$01\underline{01000 \downarrow 1}000001$	010000 was already included
$010\underline{10001 \downarrow 0}00001$	100011 is not in *L*
$0101\underline{00010 \downarrow 0}0001$	000101 was already included
$01010\underline{00100 \downarrow 0}001$	001001 is not included, we include (100)
0(10(10)0010)0(100)0001	
$010100\underline{01001 \downarrow 0}00001$	010011 is not in *L*
$0101000\underline{10010 \downarrow 0}0001$	100101 is not in *L*
$01010001\underline{00100 \downarrow 0}001$	001001 was already included
$010100010\underline{01000 \downarrow 0}01$	010001 was already included
$0101000100\underline{10000 \downarrow 0}1$	100001 is not in *L*
$01010001001\underline{00000 \downarrow 1}$	000000 is not included, we include (0)
0(10(10)0010)0(100)000(0)1	
$010100010010\underline{00000 \downarrow 1}$	000000 was already included
$01010001001000000 1 \downarrow$	end.
$\boxed{010100010010000001}$	is a de Bruijn sequence

3.7 Some cardinality results

In this section we present some cardinalities results obtained using known results in subshift of finite type.

Let $\mathcal{F}$ be a finite set of forbidden factors. Over the subshift of finite type $X_{\mathcal{F}}$, let $\mathbb{W}_k^{\mathcal{F}}(n)$ be set of periodic factors of length n simply called "words", let $\mathbb{L}_k^{\mathcal{F}}(n)$ be the set of Lyndon words of length n and let $\mathbb{N}_k^{\mathcal{F}}(n)$ be the set of bi-infinite sequence of period n simply called "necklaces". Note that the set of necklaces is the set of circular words of length n. We denote the cardinalities of these sets by $W_k^{\mathcal{F}}(n)$, $L_k^{\mathcal{F}}(n)$ and $N_k^{\mathcal{F}}(n)$ respectively.

The number of words in the language can be obtained using the *zeta function* of a subshift of finite type, this function is defined by

$$\zeta(z) = \exp\left(\sum_{i\geq 1} \frac{W_k^{\mathcal{F}}(i)}{i} z^i\right)$$

For a subshift of finite type this function can be easily calculated using the adjacency matrix M of its corresponding graph

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

and then we can use the Taylor's formula for obtain the number of words of length n in the language:

$$W_k^{\mathcal{F}}(n) = \frac{1}{(n-1)!} \frac{d^n}{dz^n} \log \zeta(z)\bigg|_{z=0} \tag{3.1}$$

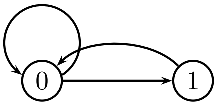


Figure 3.2: Graphical representation of the “Golden Mean” subshift.

We can also study the convergence of this number when n go to infinity, which gives the entropy of the system. The next equality is true for an irreducible subshift of finite type

$$\limsup_{n\to\infty} \frac{1}{n} \log W_k^{\mathcal{F}}(n) = h(X)$$

where $h(X)$ is the entropy of the subshift of finite type, which is equal in our case to the logarithm of the largest eigenvalue λ of the adjacency matrix. This eigenvalue is called the Perron eigenvalue of the system.

Moreover, if the subshift of finite type is irreducible and the lengths of periodical words are relatively prime (gcd$\{i : W_k^{\mathcal{F}}(i) > 0\} = 1$) then the system is *mixing*, and then we can approximate $W_k^{\mathcal{F}}(n)$ by

$$W_k^{\mathcal{F}}(n) = (1 + \rho_i(n))\,\lambda^n$$

where $\rho_i(n) \to 0$ as $n \to \infty$.

If the greatest common divisor of the number of words is not 1, there exists a similar but more complicated formula. Nevertheless, in both cases we can estimate the number of words in the language by

$$W_k^{\mathcal{F}}(n) = \Theta(\lambda^n)$$

Example 3.7.1. The Golden Mean is the subshift of finite type defined by a binary alphabet $A = \{0,1\}$ forbidding $\mathcal{F} = \{11\}$. It has a graphical representation of two states 0 and 1, and arcs from $0 \to 0$, $0 \to 1$ and $1 \to 0$ (see Figure 3.2), so its adjacency matrix is $\left[\begin{smallmatrix} 1 & 1 \\ 1 & 0 \end{smallmatrix}\right]$. Hence the entropy of the subshift of finite type is

$$h(X) = \log\left(\frac{1+\sqrt{5}}{2}\right)$$

and its zeta function is

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

Therefore, the number of words of length n is given by

$$\begin{aligned} W_k^{\mathcal{F}}(n) &= \frac{-1}{(n-1)!}\,\frac{d^{n-1}}{dz^{n-1}}\left(\frac{2z+1}{z^2+z-1}\right)\bigg|_{z=0} \\ &= \left(\frac{2}{-1+\sqrt{5}}\right)^n + \left(\frac{2}{-1-\sqrt{5}}\right)^n \\ &= \frac{(1+\sqrt{5})^n + (1-\sqrt{(}5))^n}{2^n} \end{aligned}$$

and its asymptotic behavior is given by

$$W_k^{\mathcal{F}}(n) = \Theta\left(\frac{1+\sqrt{5}}{2}\right)^n$$

The words of length n are either rotations of Lyndon words of length n or powers of rotations of Lyndon words of length d with $d|n$, so

$$W_k^{\mathcal{F}}(n) = \sum_{d|n} d \cdot L_k^{\mathcal{F}}(d)$$

applying the Möbius inversion formula we can obtain the number of Lyndon words

$$L_k^{\mathcal{F}}(n) = \frac{1}{n}\sum_{d|n}\mu(n/d)W_k^{\mathcal{F}}(d) \tag{3.2}$$

where μ is the Möbius function.

The asymptotic behavior of this number is also given by the entropy of the system

$$\limsup_{n\to\infty}\frac{1}{n}\log\left(n\cdot L_k^{\mathcal{F}}(n)\right) = h(X)$$

Also, using the fact that $L_k^{\mathcal{F}}(n) \leq (1/n)\sum_{d=1}^{n} W_k^{\mathcal{F}}(d)$ we conclude that number $L_k^{\mathcal{F}}(n) = O(\lambda^n/n)$, and as $L_k^{\mathcal{F}}(n) \geq W_k^{\mathcal{F}}(n)/n$ we can estimate the number of Lyndon words in the language by

$$L_k^{\mathcal{F}}(n) = \Theta\left(\frac{1}{n}\lambda^n\right)$$

For the number of necklaces of length n, we can use the Burnside Lemma for calculate it using the periodicity of the words

$$N_k^{\mathcal{F}}(n) = \frac{1}{n}\sum_{i=0}^{n-1} p_i$$

where p_i is the number of words of length n with period i. Any word of length n has period n, so it is easy to see that $p_i = p_{\gcd(i,n)}$. Also, if $i|n$ the words of length n with period i are the powers of the words in the language of length i, so $p_i = p_{W_k^{\mathcal{F}}(i)}$ and we obtain the following formula:

$$N_k^{\mathcal{F}}(n) = \frac{1}{n}\sum_{i=0}^{n-1} W_k^{\mathcal{F}}(\gcd(i,n))$$

Reordering the indices of the sum, we can group all indices i with $\gcd(i,n) = d$, which are exactly $\phi(n/d)$ where ϕ is the Euler function. Hence, we obtain a formula for the number of necklaces very similar to the number of Lyndon words.

$$N_k^{\mathcal{F}}(n) = \frac{1}{n}\sum_{d|n}\phi(n/d)W_k^{\mathcal{F}}(d) \tag{3.3}$$

For the asymptotic behavior, as $L_k^{\mathcal{F}}(n) \leq N_k^{\mathcal{F}}(n) = \sum_{d|n} L_k^{\mathcal{F}}(d)$, we can use the same arguments as before to prove that

$$N_k^{\mathcal{F}}(n) = \Theta\left(\frac{1}{n}\lambda^n\right)$$

Example 3.7.2. For the Golden Mean, we obtain the following values

n	1	2	3	4	5	6	7	8	9	10	11
$W_k^{\mathcal{F}}(n)$	1	3	4	7	11	18	29	47	76	123	199
$L_k^{\mathcal{F}}(n)$	1	1	1	1	2	2	4	5	8	11	18
$N_k^{\mathcal{F}}(n)$	1	2	2	3	3	5	5	8	10	15	19

Note that even for $n = 11$ these values are very close to $\left(\frac{1+\sqrt{5}}{2}\right)^n$, $\frac{1}{n}\left(\frac{1+\sqrt{5}}{2}\right)^n$ and $\frac{1}{n}\left(\frac{1+\sqrt{5}}{2}\right)^n$ respectively.

MINIMAL DE BRUIJN SEQUENCE

4.1 Introductory Notes

The BEST (de Bruijn, van Aardenne-Ehrenfest, Smith and Tutte) Theorem (see page 46) allows us to compute the number of Eulerian cycles in $G^{\mathcal{D}}$. Hence, we can compute the number of de Bruijn sequence for a dictionary $\mathcal{D}$.

This number is exponential, in fact we prove that the number of de Bruijn sequence for $\mathcal{D} = \mathcal{P}_n(X_{\mathcal{F}})$ is

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

where λ is the entropy of $X_{\mathcal{F}}$.

Therefore, to find the minimal (lexicographically) de Bruijn sequence for $\mathcal{D}$ is a difficult problem.

In the case $\mathcal{D} = A^n$, the minimal de Bruijn sequence of span n is given by the theorem of Fredricksen and Maiorana. This sequence can also be obtained has a walk over the de Bruijn graph in the following way: we start from the vertex with maximum label and we follow the unvisited arc of minimum label. By the BEST theorem we will finish with an Eulerian cycle, and by construction there is not another Eulerian cycle with a smaller label.

Based on this strategy, in Section 4.2 we study for which sets $\mathcal{D}$ the previous

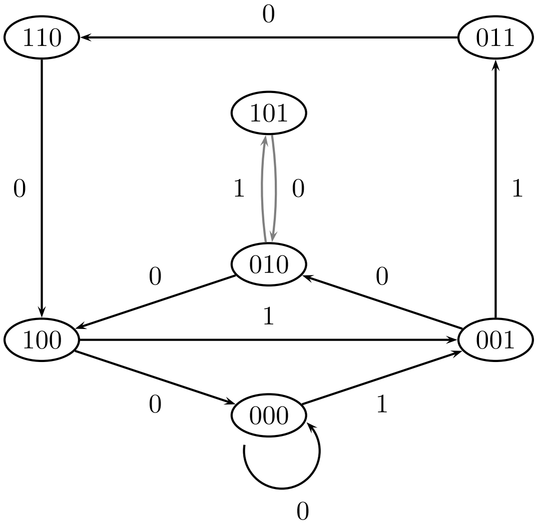


Figure 4.1: Example of de Bruijn graph for $n = 3$ and $\mathcal{F} = \{0111\}$

strategy over the graph $G^{\mathcal{D}}$ finishes with an Eulerian cycle (and therefore, with the minimal de Bruijn sequence). From this analisys we remark that there are sets $\mathcal{D}$ for which there exists a de Bruijn sequence but it can not be obtained using the previous strategy. For example, in Figure 4.1 we see an example where exists a de Bruijn sequence but previous strategy does not finish with the minimal de Bruijn sequence because the arcs in grays are not visited.

In Section 4.3 we present an algorithm in order to find the Eulerian cycle of minimal lexicographical label in a graph. This algorithm solve completely the problem of finding the minimal de Bruijn sequence. Moreover, this algorithm apply not only for de Bruijn graphs, but also for any Eulerian labeled digraph.

4.2 Full paper:

Minimal de Bruijn Sequence in a Language with Forbidden Substrings

In: Lecture Notes in Computer Science

“Proceedings of the 30th International Workshop on Graph-Theoretic Concepts in Computer Science”

Volume 3353, Pages 168–176

DOI: 10.1007/b104584

Abstract

Let be the following strategy to construct a walk in a labeled digraph: at each vertex, we follow the unvisited arc of minimum label. In this work we study for which languages, applying the previous strategy over the corresponding de Bruijn graph, we finish with an Eulerian cycle, in order to obtain the minimal de Bruijn sequence of the language.

4.2.1 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 known 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 known as "shift register sequences" and was originally studied by N. G. De Bruijn for the binary alphabet (de Bruijn, 1946). 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 (Stein, 1961; Bermond et al., 1997a; Chung et al., 1992)). Historically, de Bruijn sequence was studied in an arbitrary alphabet considering the language of all the *n*-tuples. There is a large number of de Bruijn sequence in this case, but only a few can be generated efficiently, see (Fredricksen, 1982) for a survey about this subject. In 1978, Fredricksen and Maiorana (Fredricksen and Maiorana, 1978) 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 (Ruskey et al., 1992). Recently, the study of these concepts was extended to languages with forbidden substrings: in (Ruskey and Sawada, 2000) it was given efficient algorithms to generate all the words in a language with one forbidden substring, in (Moreno, 2003) the concept of de Bruijn sequences 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 super-string problem" which is a important problem in the areas of DNA sequencing and data compression.

In this work we study the de Bruijn sequence of minimal lexicographical label. In subsection 4.2.2 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 subsections. In subsection 4.2.3 we study the main problem, giving some results on the structure of the de Bruijn graph. Finally, in subsection 4 we present some remarks and extensions of this work.

4.2.2 De Bruijn sequence of restricted languages

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 $|w|$.

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

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, $xu < yv$.

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^*. A *subshift of finite type* (SFT) is the subset of sequences in $A^{\mathbb{Z}}$ 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, in this work we will say that 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 $\mathcal{W}^{\mathcal{F}}(n)$.

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 circular string $B^{\mathcal{D}}$ of length $\left|\mathcal{W}^{\mathcal{F}}(n)\right|$ such that all the words in the language of length n are factors of

$B^{\mathcal{D}}$. In other words,

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

These concepts are studied in (Moreno, 2003), 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:

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

where $\log(\lambda)$ is the *entropy* of the system (see (Lind and Marcus, 1995)). The second result proves the existence of a de Bruijn sequence:

Theorem 4.2.1. *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 $G_n^{\mathcal{D}}$, is the largest strongly connected component of the directed graph with $|A|^n$ vertices, labeled by the words in A^n, and the set of arcs

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

where the label of the arc $e = (as, sb)$ is $l(e) = b$. Note that if the SFT is irreducible, this graph has only one strongly connected component of size greater than 1, so there is no ambiguity in the definition.

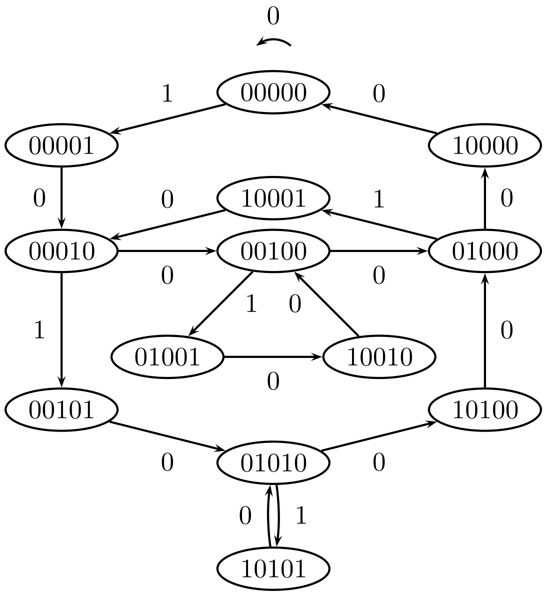


Figure 4.2: De Bruijn digraph of span 5 for the Golden Mean ($\mathcal{F} = \{11\}$)

There are not two vertices with the same label, hence from now we identify a vertex by its label. If $W = e_1 \dots e_k$ is a walk over $G_n^{\mathcal{D}}$, we denote the label of W by $l(W) = l(e_1) \dots l(e_k)$, and by $l(W)^j$ the concatenation of of j times $l(W)$.

There exists a bijection between the arcs of $G_n^{\mathcal{D}}$ and the words in $\mathcal{W}^{\mathcal{F}}(n+1)$, because to each arc with label $a \in A$ with tail at $w' \in A^n$ we can associate the word $w'a$ which is, by definition, a word in $\mathcal{W}^{\mathcal{F}}(n+1)$. Equally if $w'a$ is a word of $\mathcal{W}^{\mathcal{F}}(n+1)$, with $a \in A$, then there exists a vertex w' and an arc with tail at this vertex with label a.

Furthermore, if a word w is a label of a walk from u to v then v is a suffix of length n of uw. In the same way, if $w \in \mathcal{W}^{\mathcal{F}}(n+1)$ then there is a cycle C in $G_n^{\mathcal{D}}$ with label $l(C)$ such that $l(C)^{\frac{n+1}{|C|}} = w$.

With all these properties it is easy to see that a de Bruijn sequence of span $n+1$ is exactly the label of an Eulerian cycle over $G_n^{\mathcal{D}}$.

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 (Tutte, 1984)) 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 each vertex 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, a walk over G starting at the root and using the arc in the tree only if all the arcs with tail at this vertex has been used, is 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 also allows us 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 $((\overline{d}^+ - 1)!)^{|V|}$ where $\overline{d}^+$ 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 $2^{2^{n-1}}$.

Now, we define formally a walk "avoiding a subgraph". Let r be any vertex. For each vertex $v \neq r$ in $G_n^{\mathcal{D}}$ let e_v be any arc starting at v. Let H be the spanning subgraph of $G_n^{\mathcal{D}}$ with arc set $\{e_v : v \in V(G_n^{\mathcal{D}}) \setminus \{r\}\}$.

Is easy to see that H is composed by cycles, subtrees converging to a cycle, and one subtree converging to r. For a vertex not in a cycle of H, we define H_v as the directed subtree converging to v in H.

We define recursively a walk in $G_n^{\mathcal{D}}$ which *avoid* H. It starts at the root vertex r. Let $v_0 e_0 \cdots v_i$ be the current walk. If there is an unvisited arc $e_i = (v_i, v_{i+1})$ not in H we extend the walk by $e_i v_{i+1}$. Otherwise we use the arc e_{v_i} in H.

We say that a walk over the graph *exhausts* a vertex if the walk use all the arc having the vertex as head or tail.

The next lemma studies in which order the vertices are exhausted in a walk avoiding H

Lemma 4.2.2. *Let W be a walk starting at vertex r avoiding H, let v be a vertex and let*

Wv the subpath of W starting at vertex r and finishing when it exhausts the vertex v. Then for each vertex u in H_v, u is exhausted in Wv.

Proof. By induction in the depth of the subtree with root v. If v is a leaf of H then $H_v = \{v\}$. If v is not a leaf and Wv exhaust v, then Wv visit all arc $(v,w) \in E$, and therefore all the arcs $(u,v) \in E$, applying induction hypothesis to all vertices u such that $(u,v) \in E$ we prove the result. □

4.2.3 Minimal de Bruijn sequence

Let $m = m_1 \dots m_n$ be the vertex of $G_n^{\mathcal{D}}$ of maximum label in lexicographic order. We are interested in to obtain the Eulerian cycle of minimum label starting at m. In order to obtain this cycle, we define the following walk: Starting at m, at each vertex we continue by the arc with the lowest label between the unvisited arcs with tail at this vertex. A walk constructed by this way will be called a *minimal walk*. By definition, there is no walk with a lexicographically lower label, except its subwalks. In this subsection we characterize when a minimal walk starting at m is an Eulerian cycle, obtaining the minimal de Bruijn sequence.

For each vertex v let $e(v)$ be the arc with tail at the vertex v and with maximum label. Let T be the spanning subgraph of $G_n^{\mathcal{D}}$ composed by the set of arcs $e(v)$, for $v \in V(G_n^{\mathcal{D}})$, $v \neq m$. The label of $e(v)$ will be denoted by $\gamma(v)$.

Is easy to see that a minimal walk is a walk avoiding T, hence we can study a minimal walk by analyzing the structure of T.

Theorem 4.2.3. *A minimal walk is an Eulerian cycle if and only if T is a tree.*

Proof. A minimal walk W exhaust m, if T is a tree then by Lemma 4.2.2 all vertices of T are exhausted by W, hence W is an Eulerian cycle. Conversely, if W is an Eulerian cycle, by the BEST Theorem the subgraph composed by the last arc visited at each vertex is a tree, but this subgraph is T, concluding that T is a tree. □

In the unrestricted case (when $\mathcal{W}^{\mathcal{F}}(n) = A^n$), the subgraph T is a regular tree of depth n where each non-leaf vertex has $|A|$ sons, therefore the minimal walk is an Eulerian cycle.

In the restricted case, we do not obtain necessarily an Eulerian cycle, because T is not necessarily a spanning tree converging to the root due to the existence of cycles (see Figure 4.3).

We will study the structure of the graph $G_n^{\mathcal{D}}$ and the subgraph T, specially the cycles in T. The main theorem of this subsection characterizes the label of cycles in T, allowing us to characterize the languages where the minimal walk is an Eulerian cycle.

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

Lemma 4.2.4. *Let $k \geq n+2$. Let $W = v_0e_0v_1e_1 \cdots e_{k-1}v_k$ be a walk in T. Then $l(e_0) \leq l(e_{n+1})$.*

Proof. Since $v_n = l(e_0)\cdots l(e_{n-1})$ we have that $l(e_1)\cdots l(e_{n-1})l(e_n)l(e_0) \in \mathcal{W}^{\mathcal{F}}(n+1)$. Hence there exists an arc (v_{n+1}, u) with label $l(e_0)$, where $v_{n+1} = l(e_1)\cdots l(e_{n-1})l(e_n)$. By the definition of T, $l(e_0) \leq \gamma(v_{n+1}) = l(e_{n+1})$. □

Corollary 4.2.5. *Let C be a cycle in T. Then $|C|$ divides $n+1$. Moreover for every vertex u in C, $u\gamma(u) = l(C)^{\frac{n+1}{|C|}}$.*

Proof. Let consider the walk $W = v_0e_0 \cdots e_{|C|-1}v_{|C|} = v_0e_0 \cdots e_{(n+1)|C|-1}v_0e_0v_1$ as $n+1$ repetitions of the cycle C. From Lemma 4.2.4 we have $l(e_0) \leq l(e_{n+1}) \leq l(e_{2(n+1)}) \leq l(e_{(n+1)|C|}) = l(e_0)$. Since we can start the cycle in any vertex we conclude that $l(e_i) = l(e_{(n+1)+i})$ for every $i = 0, \dots, |C|-1$. Hence $|C|$ divides $n+1$. The second conclusion comes from the fact that the label of any walk of length at most n ending in a vertex u is a suffix of u. □

Let $u \neq m$ be a vertex. Among all the words which are prefix of m and suffix of u, let $g(u)$ be the longest one (notice that $g(u)$ could be the empty word ε and $|g(u)| < n$). Let $\alpha(u) = m_{|g(u)|+1}$ be the letter following the end of $g(u)$ in m.

Notice that in the unrestricted case, $|g(u)|$ is the distance over the graph from the vertex u to m. This function will be essential in the study of T. The next lemma give us a bound over the label of the arcs in terms of the function $g(\cdot)$.

Lemma 4.2.6. *For all pairs of adjacent vertices u and v, $l(uv) \leq \alpha(u)$. Moreover, if $l(uv) < \alpha(u)$ then $g(v) = \varepsilon$ and if $l(uv) = \alpha(u)$ then $g(v) = g(u)l(uv)$.*

Proof. $g(u)$ is a suffix of u, and $ul(uv) \in \mathcal{W}^{\mathcal{F}}(n+1)$, so $g(u)l(uv)$ is a prefix of a word in $\mathcal{W}^{\mathcal{F}}(n+1)$. Since m is the maximal word and $g(u)$ is a prefix of m we get $l(uv) \leq \alpha(u)$.

If $l(uv) = \alpha(u)$ then $g(u)l(uv)$ is a prefix of m and a suffix of v. Hence $g(u)l(uv)$ is a suffix of $g(v)$. Since by removing the last letter of a suffix of v we obtain a suffix of u we conclude $g(v) = g(u)l(uv)$.

We show that if $g(v) \neq \varepsilon$ then $\alpha(u) \geq l(uv)$. Let $g(v) = g'(v)l(uv)$, then $g'(v)$ is a suffix of u and a prefix of m. Hence $g'(v)$ is a suffix of $g(u)$. Therefore $g'(v)\alpha(u)$

is a factor of m. By the definition of $g(v)$ and the maximality of m $g(v)$ is greater or equal (lexicographically) than $g'(v)\alpha(u)$. We conclude that $\alpha(u) \geq l(uv)$. □

In the unrestricted case, where T is a tree of depth n, all the arcs not in T go to a leaf. In the general case we can define an analog to the leaves.

We say that a vertex u is a *floor* vertex if $g(u) = \varepsilon$. Notice that in the unrestricted case the leaves of T are the floor vertices. We say that a vertex u is a *restricted* vertex if $\gamma(u) < \alpha(u)$.

Corollary 4.2.7. *If a cycle in T contains l restricted vertices, then it has exactly l floor vertices.*

Proof. From Lemma 4.2.6 we know that if a vertex u is restricted then for every arc (u,v) the vertex v is a floor vertex. To conclude it is enough to see that in T an arc (u,v) with u unrestricted has label $\alpha(u)$. Then v is not a floor vertex. □

Corollary 4.2.8. *Let P be a path in T starting in a floor vertex, ending in a vertex v and with unrestricted inner vertices. Then $l(P) = g(v)$.*

Proof. We apply induction on the length of P. The case where the length of P is zero is direct since v is a floor vertex. Let us consider the case where P has length at least 1. Since v is not a restricted vertex, from Lemma 4.2.6 we know that $g(v) = g(u)l(uv)$, where u is its neighbor in P. By the induction assumption $g(u) = l(P')$ where P' is the path obtained from P removing the arc (u,v). Hence $g(v) = l(P')l(uv) = l(P)$. □

We will use these results to characterize the label of cycles in T, specially we will characterize the restricted vertices of a cycle.

Theorem 4.2.9. *Let C be a cycle in T, let $u^0,\dots,u^{k-1}$ be the restricted vertices in C ordered according to the order of C. Then $u^i = g(u^{i+1})\gamma(u^{i+1})\cdots\gamma(u^{i-1})g(u^i)$ for $i = 0,\dots,k-1$, where $i+1,\dots,i-1$ are computed* $\mod k$.

Proof. From Corollary 4.2.8 the label of C is $g(u^0)\gamma(u^0)\cdots\ g(u^{k-1})\ \gamma(u^{k-1})$, and by definition of $G_n^{\mathcal{D}}$, u^i is the label of any walk over $G_n^{\mathcal{D}}$ of length n finishing in u^i, so by Corollary 4.2.5 we can take the walk C^k composed by $k = (n+1)/|C|$ repetitions of C finishing in u^i, concluding that $u^i = g(u^{i+1})\gamma(u^{i+1})\cdots\gamma(u^l)\ l(C^{k-1})g(u_1)\ \cdots\ \gamma(u^{i-1})\ g(u^i)$. □

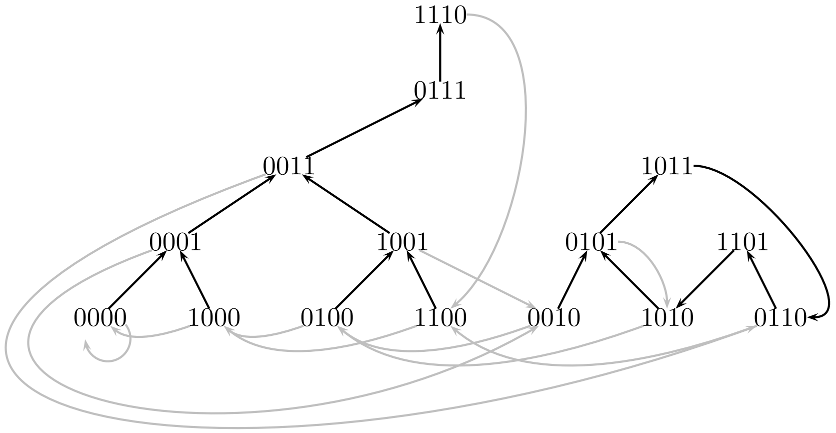


Figure 4.3: Example of the subgraph T for $n = 4$ and $\mathcal{F} = \{01111\}$ in a binary alphabet.

Now we are able to give a characterization of the languages where a minimal walk produces an Eulerian cycle.

Let $\mathcal{H}$ be the subset of $\mathcal{W}^{\mathcal{F}}(n+1)$ where $w \in \mathcal{H}$ if and only if w can be decomposed by $w = h^0\beta_1\dots h^{k-1}\beta_{k-1}$ where each $h^i \in A^*$ and $\beta_i \in A$ satisfy the following conditions:

1. $h^i = m_1\dots m_{|h^i|}$ (a prefix of m)

2. $\beta_i < m_{|h^i|+1}$

3. $\forall \beta' > \beta_i,\ h^{i+1}\beta_{i+1}\dots\beta_{i-1}h^i\beta' \notin \mathcal{W}^{\mathcal{F}}(n+1)$

Now, we are able to characterize the languages where a minimal walk is an Eulerian cycle.

Theorem 4.2.10. *A minimal walk is an Eulerian cycle if and only if $\mathcal{H} = \emptyset$.*

Proof. From Theorem 4.2.3, we have to prove that T is a tree if and only if $\mathcal{H} = \emptyset$.

If T is not a tree then T has a cycle C. Let $u^0 \dots u^{k-1}$ be the restricted vertices of the cycle. By Theorem 4.2.9 $l(C) = g(u^0)\gamma(u^0)\dots g(u^{k-1})\gamma(u^{k-1})$ and by Corollary 4.2.5 $|C|$ divides $n+1$. Therefore there exists a word w in $\mathcal{W}^{\mathcal{F}}(n+1)$ composed by $(n+1)/|C|$ repetitions of C. By definition of $\mathcal{H}$ we conclude that $w \in \mathcal{H}$.

Conversely, let us assume that T has no cycles and $\mathcal{H} \neq \emptyset$. Let w be a word in $\mathcal{H}$. By definition of $G_n^{\mathcal{D}}$, there is a cycle C in $G_n^{\mathcal{D}}$ of length dividing $n+1$ such that C (or repetitions of C) has label w. We shall prove that C is also a cycle in T.

Let v be a vertex of C, with $v = \dots\beta_{i-1}(h^i)_1\dots(h^i)_j$ where $j = 0\dots|h^i|$. If $0 < j < |h^i|$, then $m_1\dots m_j$ is a suffix of v, so $\alpha(v) = m_{j+1} = (h^i)_{j+1}$ hence the arc of C with tail at v is in T. If $j = 0$ then $\gamma(v) = m_1$ therefore the arc in C is in T. Finally, let consider the case $j = |h^i|$.If (v, v') is the arc in C then $l(vv') = \beta_i$. Since $w \in \mathcal{H}$, no arc in $G_n^{\mathcal{D}}$ with tail at v has a label greater than β_i. Then $(v, v') \in T$. We conclude that C is a cycle in T which leads to a contradiction. □

4.2.4 Some remarks

The previous analysis considers only the minimal walk starting at the root vertex. This case does not necessarily produce the minimal label over all Eulerian cycles, because there can be Eulerian cycles starting at a non root vertex with a lexicographically lower label.

It is also possible to construct an algorithm which modifies T in order to destroy cycles in T, and obtain the minimal de Bruijn sequence for any irreducible subshift of finite type. However further research in this subject allow us to construct an algorithm to obtain the minimal Eulerian cycle for any edge-labeled digraph (see (Matamala and Moreno, 2004)). This algorithm is presented in the next section.

4.3 Full paper:

Minimal Eulerian cycle in a labeled digraph

Tech. Report. CMM-B-04/08-108

DIM-CMM, Universidad de Chile, August 2004.

ArXiv: cs.DM/0505036

Abstract

Let G be an Eulerian directed graph with an arc-labeling such that arcs going out from the same vertex have different labels. In this work, we present an algorithm to construct the Eulerian trail starting at an arbitrary vertex v of minimum lexicographical label among labels of all Eulerian trails starting at this vertex.

We also show an application of this algorithm to construct the minimal de Bruijn sequence of a language.

4.4 Introduction

Eulerian graphs were an important concept in the beginning of the graph theory. The “Königsberg bridge problem” and its solution given by Euler in 1736 are considered the first paper of what is nowadays called *graph theory*.

In this work, we consider graphs with an arc-labeling with the following property: Arcs going out from the same vertex have different labels. These graphs are commonly utilized in the automata theory: a labeled digraph represents deterministic automata where vertices are the states of the automata, and arcs represent the transitions from one state to another, depending on the label of the arc. Eulerian trails over these graphs are related with synchronization of automata (see (Kari, 2003)).

Eulerian graphs with this kind of labeling are also used in the study of DNA. By DNA sequencing we can obtain fragments of DNA which need to be assembled in the correct way. To solve this problem, we can simply construct a *DNA graphs* (see (Blazewicz et al., 1999)) and find an Eulerian trail over this graph. This strategy is already implemented and it is now one of the more promising algorithms for DNA sequencing (see (Pevzner, 1989; Pevzner et al., 2001)).

To find the Eulerian trail of minimal label is also interesting with respect to the problem of finding of optimal encoding for DRAM address bus. In this model, an address space of size 2^{2n} is represented as labels of edges in a complete graph with 2^n vertices. An Eulerian trail over this graph produces an optimal multiplexed code (see (Cheng and Pedram, 2000)). If we want to give priority to some address in particular, the Eulerian cycle of minimal label give us this code.

Another interesting application of these graphs is to find *de Bruijn sequences* of a language. De Bruijn sequences are also known as "shift register sequences" and were originally studied in (de Bruijn, 1946) by N. G. De Bruijn for the binary alphabet. These sequences have many different applications, such as memory wheels in computers and other technological device, network models, DNA algorithms, pseudorandom number generation and modern public-key cryptographic schemes, to mention a few (see (Stein, 1961; Bermond et al., 1997a; Chung et al., 1992)). More details about this application are discussed in Section 3.

By the BEST theorem (see (Tutte, 1984)), we can compute the number of Eulerian trails in a graph. This number is usually exponential in the number of vertices of the graph (at least $((\gamma-1)!)^{|V|}$ where V is the set of vertices and γ is the minimum degree of vertices in V). Therefore, finding the Eulerian trail of lexicographically minimum label can be costly.

In this work, we give an algorithm to construct the Eulerian trail of minimum label starting at a given vertex. The complexity of the algorithm is linear in the number of arcs of the graph. In Section 2 we give some definitions to understand the problem and we prove the main theorem. Finally, in Section 3 we give an application of this algorithm to construct the minimal de Bruijn sequence of a language.

4.5 Main Theorem

Let G be a digraph and let $l : A(G) \to N$ be a labeling of the arcs of G over an alphabet N such that arcs going out from the same vertex have different labels.

A *trail* is an alternating sequence $W = v_1 a_1 v_2 a_2 \ldots v_{k-1} a_{k-1} v_k$ of vertices v_i and arcs a_j such that the tail of a_i is v_i and the head of a_i is v_{i+1} for every $i = 1, 2, \ldots, k-1$ and all arcs are distinct. If $v_1 = v_k$ then W is a closed trail. A closed trail is an Eulerian trail if the arcs of W are all the arcs of G. An Eulerian graph is a graph with an Eulerian trail. The label of W is the word $l(a_1) \ldots l(a_{k-1})$.

Given a strongly connected Eulerian digraph and a vertex r, we show how to find the Eulerian trail starting in r with the minimal lexicographical label. Remark that is important to fix a starting vertex r so at to define an order in which vertices are visited, which allow us to define a lexicographical order among Eulerian trails.

Let U be a subset of vertices in G. A *cut* is the set of arcs with one end in U and the other in $V \setminus U$, and is denoted by $\delta_G(U)$. A vertex v is *exhausted* by a trail W if $\delta_{G \setminus A(W)}(v) = \emptyset$. The set of vertices exhausted by W is denoted by $S(W)$.

Lemma 4.5.1. *Let U, B be subsets of vertices and let T be the trail starting in r of minimum* *label exhausting U. If $U \subseteq B \subseteq S(T)$, then T is the trail of minimum label exhausting B.*

Proof. Let T' be a trail starting in r exhausting B with a smaller label than T. Since $U \subseteq B$ then T' exhausts U. Hence, the label of T is not minimal. □

A trail W can visit a vertex v many times. We decompose a trail W in the sub-trails Wv and vW, where Wv is the sub-trail of W finishing in the *last* visit of v, and vW is the sub-trail of W starting from the *last* visit of v. We denote $\overset{\circ}{v}W$ the trail vW without the first vertex v but containing the first arc of vW.

Lemma 4.5.2. *Let T be a closed trail starting in r such that $r \in S(T)$. Let v be the last vertex not belonging to $S(T)$ visited by T. If w is its next vertex in T then*

$$\delta_{G\setminus A(Tv)}(\overset{\circ}{v}T) = \{vw\}$$

Proof. Let xy be an arc of $\delta_G(\overset{\circ}{v}T)$. Since all vertices of $\overset{\circ}{v}T$ are exhausted by T, $xy \in A(T)$. Hence either $xy \in A(Tv)$ or $xy \in A(vT)$. Therefore $xy \in \delta_{G\setminus A(Tv)}(\overset{\circ}{v}T)$ if and only if $xy = vw$. □

We define the following strategy to construct a trail: Starting at a given vertex v, follow the unvisited arc (if exists) of minimal label. This strategy finishes with a closed trail, and this trail exhausts the vertex v. A trail constructed by this strategy is called an *alphabetic trail* starting at v and is denoted by $W(G,v)$. By definition, an alphabetic trail starting at v is the trail of minimal label among all trails starting at v and exhausting v.

Let v be a vertex and let T be the closed trail of minimal label exhausting all vertices in $\mathring{v}T$. We find the trail of minimal label exhausting all vertices in vT. If $v \in S(T)$ then by Lemma 4.5.1 the trail T is the solution to this problem. If $v \notin S(T)$ then the next lemma give us the solution: we need to split T and insert the alphabetic trail over $G \setminus A(T)$ starting at v. Repeating this process we finish with the Eulerian trail of minimal label.

Lemma 4.5.3. *Let T be a closed trail starting and exhausting r such that if v is the last vertex in $V \setminus S(T)$ visited by T then T is the closed trail of minimum label exhausting $\mathring{v}T$.*

Let Z be the closed trail of minimum label in exhausting vT and let $W = W(G \setminus A(T), v)$. Then $Z = (Tv)W(vT)$.

Proof. By supposition, T is the closed trail of minimum label exhausting $\mathring{v}T$ and $\mathring{v}T \subset S(Z)$, hence $l(Z) \geq l(T)$. In particular, $l(Z) \geq l(Tv)$. Also Z and $(Tv)W(vT)$ exhausts vT. Hence $l(Z) \leq l((Tv)W(vT))$, concluding that $Z = (Tv)Z'$.

By Lemma 4.5.2 the only way to visit vertices in $\mathring{v}T$ is using the arc vw, and $\mathring{v}T$ is the trail of minimum label exhausting $V(\mathring{v}T)$ in $G \setminus (A(Tv))$. Since Z is a closed trail of minimum label, $Z = (Tv)Z''(vT)$.

Finally, Z'' is a closed trail of minimum label in $G \setminus A(T)$ exhausting v, therefore $Z'' = W$. □

Theorem 4.5.4. *Algorithm 4.5.1 finishes with an Eulerian trail starting in r and its label is the minimal one among all Eulerian trails starting in r.*

Proof. At each repetition of the “while”, the trail T exhausts at least one vertex non-exhausted in the previous step, so the algorithm finishes in a finite number of steps.

Algorithm 4.5.1: Compute the minimal Eulerian trail starting in r

```
$T \leftarrow \emptyset$
$v \leftarrow$NoEx$(T)$  $\{v = r\}$
**while** $v \neq$ **NULL do**
   $W \leftarrow W(G \setminus A(T), v)$.over $G \setminus A(T)$.
   $T \leftarrow (Tv)W(vT)$
   $v \leftarrow$ NoEx$(T)$.
**end while**
```

Where NoEx(T) returns the last non-exhausted vertex visited by T or **NULL** if this vertex does not exist.

We define inductively $G^i = G \setminus A(T^{i-1})$, $v^i =$NoEx(T_i), $W^i = W(G^i, v^i)$ and $T^i = (T^{i-1}v^{i-1})W^i(v^{i-1}T^{i-1})$, with $T_0 = \emptyset$.

We prove by induction that T^i is the closed trail of minimal label exhausting $\overset{\circ}{v}^{i}T^i$. For $i = 1$, $T^1 = W(G, r)$ is by definition the closed trail of minimal label exhausting r, and by Lemma 4.5.1 it is the trail of minimal label exhausting $\overset{\circ}{v}^{1}T^1$. Let T^{i-1} be the closed trail of minimum label exhausting $\overset{\circ}{v}^{i-1}T^{i-1}$. Applying Lemma 4.5.3 to T^{i-1}, we conclude that T^i is the closed trail of minimal label exhausting $v^{i-1}T^i$ and by Lemma 4.5.1 it is the minimal closed trail exhausting $\overset{\circ}{v}^{i}T^i$.

Therefore the algorithm finishes with a closed trail T exhausting all its vertices $V(T)$, but G has only one strongly connected component, thus $A(T) = A(G)$. We conclude that T is an Eulerian trail of minimal label. □ □

We can use the following structure to represent the graph, a list of size $|V|$ representing vertices where each element v in the list has a stack with the head of each arc starting at v in order. Knowing this structure of a graph, the algorithm can easily construct the trails $W(\cdot,\cdot)$, removing the visited nodes from the stack and keeping track of exhausted vertices. Since this algorithm visits each arc at most twice, it can be implemented in

$O(|A(G)|)$, which is best possible.

Remark that while the initial vertex r can be arbitrarily chosen, different initial vertices can produce different trails, even if we consider the label as a circular string. For example, in the graph of Figure 4.4, the minimal de Bruijn sequence starting at u is 001122 but starting at v is 100122.

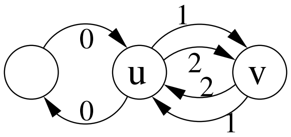


Figure 4.4:

4.6 An application: minimal de Bruijn sequence

Given a set $\mathcal{D}$ of words of length n, a de Bruijn sequence of span n is a periodic sequence B such that every word in $\mathcal{D}$ (and no other n-tuple) appears exactly once in B. Historically, de Bruijn sequence was studied in an arbitrary alphabet considering the language of all the n-tuples. In (Moreno, ears) the concept of de Bruijn sequences 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. Nevertheless, it remained to find the minimal de Bruijn sequence in this general case.

In (Moreno and Matamala, 2004) was studied some particular cases where it is possible to obtain efficiently the minimal de Bruijn sequence. Using our previous algorithm we can solve this problem efficiently in all cases.

A word p is said to be a *factor* of a word w if there exist words $u,v \in N^*$

such that $w = upv$. If u is the empty word (denoted by ε), then p is called a *prefix* of w, and if v is empty then is called a *suffix* of w.

Let $\mathcal{D}$ be a set of words of length $n+1$. We call this set a *dictionary*. A *de Bruijn sequence of span n* $+1$ for $\mathcal{D}$ is a (circular) word $B^{\mathcal{D},n+1}$ of length $|\mathcal{D}|$ such that all the words in $\mathcal{D}$ are factors of $B^{\mathcal{D},n+1}$. In other words,

$$\{(B^{\mathcal{D},n+1})_i \ldots (B^{\mathcal{D},n+1})_{i+n \mod (n+1)} | i = 0 \ldots |\mathcal{D}|\} = \mathcal{D}$$

De Bruijn sequences are closely related to de Bruijn graphs. The *de Bruijn graph of span n*, denoted by $G^{\mathcal{D},n}$, is the directed graph with vertex set

$$V(G^{\mathcal{D},n}) = \{u \in N^n | u \text{ is a prefix or a suffix of a word in } \mathcal{D}\}$$

and arc set

$$A(G^{\mathcal{D},n}) = \{(\alpha v, v\beta) | \alpha, \beta \in N, \alpha v \beta \in \mathcal{D}\}$$

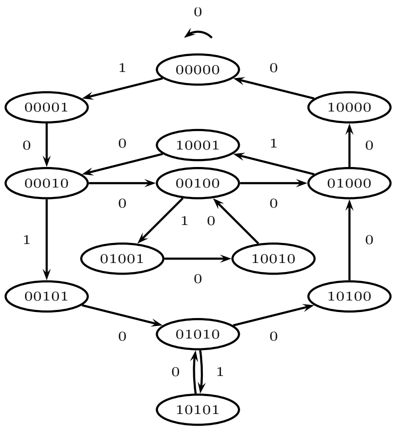
Note that the original definitions of de Bruijn sequences and de Bruijn graph given in (de Bruijn, 1946) are the particular case of $\mathcal{D} = N^{n+1}$.

We label the arcs of the graph $G^{\mathcal{D},n}$ using the following function l: if $e = (\alpha u, u\beta)$ then $l(e) = \beta$. This labeling has an interesting property: Let $T = v_0 e_0 \ldots e_m v_{m+1}$ be a trail over $G^{\mathcal{D},n}$ of length $m \geq n$. Then T finishes in a vertex u if and only if u is a suffix of $l(T) = l(e_0) \ldots l(e_m)$. This property explains the relation between de Bruijn graphs and de Bruijn sequence: $B^{\mathcal{D},n+1}$ is the label of an Eulerian trail of $G^{\mathcal{D},n}$. Therefore, given a dictionary $\mathcal{D}$, the existence of a de Bruijn sequence of span $n+1$ is characterized by the existence of an Eulerian trail over $G^{\mathcal{D},n}$.

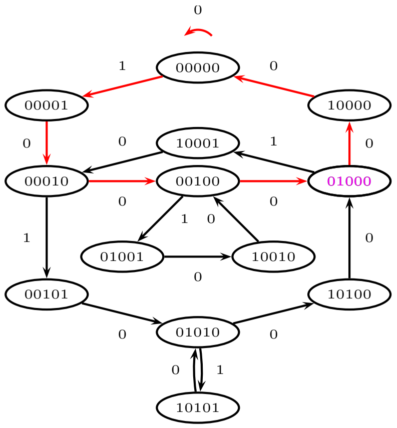
Let D be a dictionary such that $G^{\mathcal{D},n}$ is an Eulerian graph. Let z be the vertex of minimum label among all vertices. Clearly, the minimal de Bruijn sequence has z as prefix. Hence, the minimal Eulerian trail over $G^{\mathcal{D},n}$ starts at an (unknown) vertex and after n steps it arrives to z. Therefore if we start our Algorithm 4.5.1 in the vertex z we obtain the Eulerian trail of minimal label starting at z which have label $B = B' \cdot z$. Hence $z \cdot B'$ is the minimal de Bruijn sequence of span $n+1$ for $\mathcal{D}$.

Example 4.6.1. Example of the algorithm for the Golden Mean system ($\mathcal{F} = \{11\}$) with length $n = 6$.

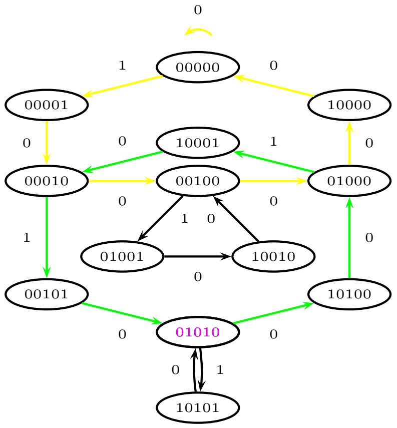
0



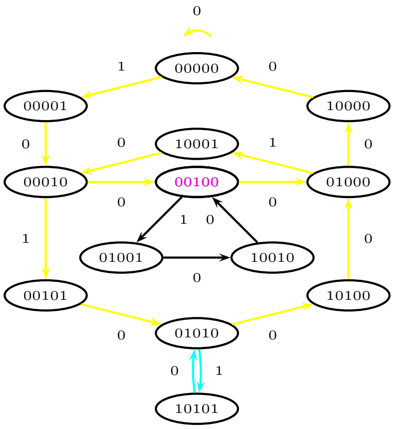
0100000



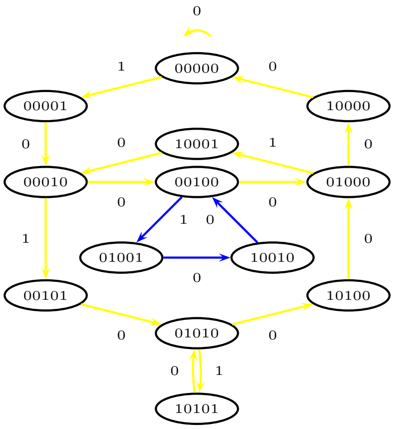
0100010100000



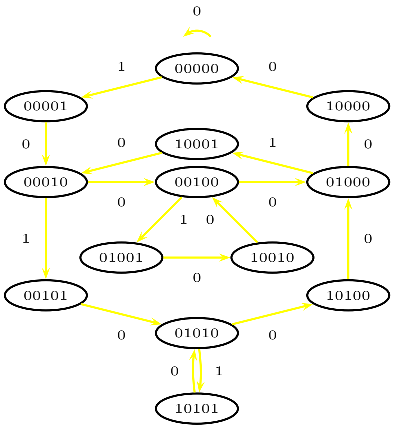
010001010100000



010010001010100000



000000100100010101



CONCLUSIONS

The main contribution of this thesis to the area of combinatorics on words are, on one hand the generalization of the concept of de Bruijn sequences, and on the other hand, the mixture of three areas of mathematics (combinatorics on words, graph theory and symbolic dynamics) used to carry out the study of this problem from different points of view.

In the area of graph theory, the main result is the Algorithm 4.5.1 allowing us to obtain in linear time the Eulerian trail of minimal label for any Eulerian digraph. An interesting open question is to generalize this algorithm to undirected graphs.

In the area of symbolic dynamics, the representation of a subshift of finite type by a de Bruijn graph opens a lot of questions. We study in depth a subset of dictionaries defined by periodic words of a subshift. However the set of dictionaries with a de Bruijn sequence (characterized in Corollary 3.4.5) is larger. It remains to give a dynamical interpretation to this set of dictionaries. Other question is about the transformation of the graphical minimal representation of a subshift of finite type to a de Bruijn (or simply Eulerian) graph. It is interesting to study if it is possible to obtain this transformation using the *state splitting* operations.Recent research in this problem shows that we can obtain the de Bruijn graph for $\mathcal{L}_n(X_{\mathcal{F}})$ using states splitting operation over the de Bruijn graph for $\mathcal{L}_{n-1}(X_{\mathcal{F}})$, leading us to think that a general transformation is possible.

Finally, the main contributions of this thesis in the area of combinatorics of words are the study of connections between the minimal de Bruijn sequences and Lyndon words. They are completely explained by Corollary 3.6.2 and the results of Chapter 4. On top of that, the generalization of de Bruijn sequence to any subset of words is an important step in the problem of generate all the n-tuples of a language. It remains to generalize the results in order to generate all the words, necklaces and Lyndon words of a language with forbidden factors. This problem can be attacked using developments in the area of "pattern matching".

We expect to open the scope of these problems in order to continue the understanding of these combinatorial objects.

Referencias

Bermond, J.-C., Dawes, R. W., and Ergincan, F. Ö. (1997a). De Bruijn and Kautz bus networks. *Networks*, 30(3):205–218. 1, 1, 3.1, 3.3, 4.2.1, 4.4

Bermond, J.-C., Liu, Z., and Syska, M. (1997b). Mean eccentricities of de Bruijn networks. *Networks*, 30(3):187–203. 1, 3.1

Berstel, J. and Karhumäki, J. (2003). Combinatorics on words—a tutorial. *Bull. Eur. Assoc. Theor. Comput. Sci. EATCS*, (79):178–228. 1

Berstel, J. and Pocchiola, M. (1994). Average cost of Duval's algorithm for generating Lyndon words. *Theoret. Comput. Sci.*, 132(1-2):415–425. 1, 2.1

Blazewicz, J., Hertz, A., Kobler, D., and de Werra, D. (1999). On some properties of DNA graphs. *Discrete Appl. Math.*, 98(1-2):1–19. 4.4

Bryant, R. D. and Fredricksen, H. (1991). Covering the de Bruijn graph. *Discrete Math.*, 89(2):133–148. 1, 3.1

Cassaigne, J. (1997). Complexité et facteurs spéciaux. *Bull. Belg. Math. Soc.*, 4:67–88. 3.5

Cattell, K., Ruskey, F., Sawada, J., Serra, M., and Miers, C. R. (2000). Fast algorithms to generate necklaces, unlabeled necklaces, and irreducible polynomials over GF(2). *J. Algorithms*, 37(2):267–282. 2.1

Cheng, W.-C. and Pedram, M. (2000). Power-optimal encoding fod DRAM address bus. In *ISLPED*, pages 250–252. ACM. 4.4

Chung, F., Diaconis, P., and Graham, R. (1992). Universal cycles for combinatorial structures. *Discrete Math.*, 110(1-3):43–59. 1, 2.1, 3.3, 4.2.1, 4.4

Crochemore, M., Hancart, C., and Lecroq, T. (2001). *Algorithmique du texte*. Vuibert. 3.6

de Bruijn, N. G. (1946). A combinatorial problem. *Nederl. Akad. Wetensch., Proc.*, 49:758–764. 1, 3.1, 3.3, 3.4, 4.2.1, 4.4, 4.6

de Bruijn, N. G. and Erdös, P. (1948). On a combinatorial problem. *Nederl. Akad. Wetensch., Proc.*, 51:1277–1279. Also: Indagationes Math. **10, 421–423 (1948)**. 2.1

Du, D. Z. and Hwang, F. K. (1988). Generalized de Bruijn digraphs. *Networks*, 18(1):27–38. 1

Duval, J.-P. (1983). Factorizing words over an ordered alphabet. *J. Algorithms*, 4:363–381. 1

Etzion, T. (1986). An algorithm for generating shift-register cycles. *Theoret. Comput. Sci.*, 44(2):209–224. 1

Flye Sainte-Marie, C. (1894). Question 48. *L’Intermédiaire Math.*, 1:107–110. 3.1, 3.4

Fredricksen, H. (1982). A survey of full length nonlinear shift register cycle algorithms. *SIAM Rev.*, 24(2):195–221. 1, 2.2.1, 3.6, 4.2.1

Fredricksen, H. (1992). A new look at the de Bruijn graph. *Discrete Appl. Math.*, 37/38:193–203. 1, 3.1

Fredricksen, H. and Kessler, I. (1977). Lexicographic compositions and deBruijn sequences. *J. Combinatorial Theory Ser. A*, 22(1):17–30. 2.1

Fredricksen, H. and Maiorana, J. (1978). Necklaces of beads in k colors and k-ary de Bruijn sequences. *Discrete Math.*, 23:207–210. 1, 2.1, 2.2.1, 3.6, 4.2.1

Golomb, S. W. (1967). *Shift register sequences*. With portions co-authored by Lloyd R. Welch, Richard M. Goldstein, and Alfred W. Hales. Holden-Day Inc., San Francisco, Calif. 1

Good, I. J. (1946). Normal recurring decimals. *J. London Math. Soc.*, 21:167–169. 3.1, 3.4

Gray, F. (1958). Pulse code communication. US Patent 2,632,058. 1

Hauge, E. R. and Mykkeltveit, J. (1996). On the classification of de Bruijn sequences. *Discrete Math.*, 148(1-3):65–83. 1, 2.1

Kari, J. (2003). Synchronizing finite automata on Eulerian digraphs. *Theoret. Comput. Sci.*, 295(1-3):223–232. Mathematical foundations of computer science (Mariánské Lázně, 2001). 4.4

Knuth, D. E. (to appear). *The Art of Computer Programming*, volume 4. Addison Wesley. Pre-fascicle 2a, Section 7.2.1.1. 1

Lind, D. and Marcus, B. (1995). *Symbolic Dynamics and Codings*. Cambridge University Press. 3.5, 4.2.2

Lothaire, M. (1997). *Combinatorics on words*. Cambridge University Press. 1

Lothaire, M. (2002). *Algebraic Combinatorics on words*. Cambridge University Press. 1

Lothaire, M. (to appear.). Applied combinatorics on words. 1

Lu, C., Xu, J., and Zhang, K. (2000). On $(d,2)$-dominating numbers of binary undirected de Bruijn graphs. *Discrete Appl. Math.*, 105(1-3):137–145. 1, 3.1

Martin, M. (1934). A problem in arrangements. *Bull. Amer. Math. Soc.*, 40:859864. 2.1

Matamala, M. and Moreno, E. (2004). Minimal Eulerian cycle in a labeled digraph. Technical Report CMM-B-04/08-108, DIM-CMM, Universidad de Chile. 4.2.4

Mitchell, C. J., Etzion, T., and Paterson, K. G. (1996). A method for constructing decodable de Bruijn sequences. *IEEE Trans. Inform. Theory*, 42(5):1472–1478. 1, 2.1

Moreno, E. (2003). Lyndon words and de Bruijn sequences in a subshift of finite type. In *Proceedings of WORDS'03*, volume 27 of *TUCS Gen. Publ.*, pages 400–410. Turku Cent. Comput. Sci., Turku. 4.2.1, 4.2.2

Moreno, E. (to appears). De Bruijn sequences and de Bruijn graphs for a general language. *Inf. Process. Lett.* 4.6

Moreno, E. and Matamala, M. (2004). Minimal de Bruijn sequence in a language with forbidden subtrings. In Hromkovic, J., Nagl, M., and Westfechtel, B., editors, *Graph-Theoretic Concepts in Computer Science*, volume 3353 of *Lect. Notes in Comp. Sci.*, pages 168–176. Springer-Verlag Heidelberg. 4.6

Mykkeltveit, J. (1972). A proof of Golomb's conjecture for the de Bruijn graph. *J. Combinatorial Theory Ser. B*, 13:40–45. 3.1

Pendavingh, R., Schuurman, P., and Woeginger, G. J. (2001). de Bruijn graphs and DNA graphs (extended abstract). 2204:296–305. 1

Pevzner, P. A. (1989). L-tuple DNA sequencing: computer analysis. *J. Biomol. Struct. Dyn.*, 7:63–73. 4.4

Pevzner, P. A., Tang, H., and Waterman, M. S. (2001). An eulerian path approach to DNA fragment assembly. *Proceedings of the National Academy of Sciences*, 98(17):9748–9753. 4.4

Ralston, A. (1981). A new memoryless algorithm for de Bruijn sequences. *J. Algorithms*, 2(1):50–62. 1

Rauzy, G. (1983). Suites à termes dans un alphabet fini. In *Seminar on number theory, 1982–1983 (Talence, 1982/1983)*, pages Exp. No. 25, 16. Univ. Bordeaux I, Talence. 3.4

Ruskey, F., Savage, C., and Wang, T. M. (1992). Generating necklaces. *J. Algorithms*, 13(3):414–430. 1, 2.1, 3.6, 4.2.1

Ruskey, F. and Sawada, J. (1999). An efficient algorithm for generating necklaces with fixed density. *SIAM J. Comput.*, 29(2):671–684 (electronic). 2.1

Ruskey, F. and Sawada, J. (2000). Generating necklaces and strings with forbidden substrings. *Lect. Notes Comput. Sci.*, 1858:330–339. 3.6, 4.2.1

Savage, C. (1997). A survey of combinatorial Gray codes. *SIAM Rev.*, 39(4):605–629. 1

Sawada, J. (2001). Generating bracelets in constant amortized time. *SIAM J. Comput.*, 31(1):259–268 (electronic). 2.1

Sawada, J. (2003). A fast algorithm to generate necklaces with fixed content. *Theoret. Comput. Sci.*, 301(1-3):477–489. 2.1

Sawada, J. and Ruskey, F. (2003). Generating Lyndon brackets. An addendum to: “Fast algorithms to generate necklaces, unlabeled necklaces and irreducible polynomials over GF(2)” [J. Algorithms **37** (2000), no. 2, 267–282; MR 2002a:05006] by K. Cattell, Ruskey, Sawada, M. Serra and C. R. Miers. *J. Algorithms*, 46(1):21–26. 2.1

Shiloach, Y. (1981). Fast canonization of circular strings. *J. Algorithms*, 2(2):107–121. 2.1

Stein, S. K. (1961). The mathematician as an explorer. *Sci. Amer.*, 204(5):148–158. 1, 2.1, 3.3, 4.2.1, 4.4

Tuliani, J. (2001). de Bruijn sequences with efficient decoding algorithms. *Discrete Math.*, 226(1-3):313–336. 1, 2.1

Tutte, W. T. (1984). *Graph theory*, volume 21 of *Encyclopedia of Mathematics and its Applications*. Addison-Wesley Publishing Company Advanced Book Program, Reading, MA. 4.2.2, 4.4