Minimal Eulerian trail in a labeled digraph

Martín Matamala [a] and Eduardo Moreno [a]

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

Abstract

Let G be an Eulerian directed graph with an arc-labeling. In this work we present an algorithm to construct the Eulerian trail starting at a 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.

Key words: Eulerian graphs, labeled digraph, de Bruijn sequence

1 Introduction

Eulerian graphs are an important concept in the beginning of the graph theory. The "Königsberg bridge problem" and its solution given by Euler in 1736 is 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 deterministic automata is represented by a labeled digraph where vertices are the states of the automata, and arcs represents the transition from one state to another, depending on the label of the arc. Eulerian trails over these graphs are related with synchronization of automatas (see [1]).

Email addresses: `mmatamal@dim.uchile.cl` (Martín Matamala), `emoreno@dim.uchile.cl` (Eduardo Moreno).

[1] Partially supported by ECOS C00E03 (French-Chilean Cooperation), Programa Iniciativa Científica Milenio P01-005, and CONICYT Ph.D. Fellowship.

Eulerian graphs with these kind of labellings are also used in the study of DNA. By DNA sequencing we 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 [2]) and find an Eulerian trail. This strategy is already implemented and is now one of the more promising algorithm for DNA sequencing (see [3,4]).

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 [5] 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, pseudo-random number generation, modern public-key cryptographic schemes, to mention a few (see [6–8]). More details about this application will be discussed in section 3.

By the BEST theorem (see [9]) 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, to find the Eulerian trail of lexicographically minimum label can be a costly problem.

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.

2 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 no two arcs going out from a vertex have the same label.

A *trail* is an alternating sequence $W = v_1a_1v_2a_2 \dots v_{k-1}a_{k-1}v_k$ 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 $i = 1, 2, \dots, k-1$ 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 the arcs of G. An Eulerian graph is a graph with an Eulerian trail. The label of W is the word $l(a_1) \dots l(a_{k-1})$.

We will show how to find in an strongly connected Eulerian digraph the Eulerian trail starting at a particular vertex r with the minimal lexicographical label.

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

Lemma 1 *Let T be the trail of minimum label among all trails exhausting a set of vertices A. Let $B \supseteq A$ be a set of vertices contained in the set of vertices exhausted by T. Then T is the trail of minimum label among all trails exhausting B.*

PROOF. Let T' be a trail exhausting B with a smaller label than T. Since $A \subseteq B$ then T' exhaust A. Hence the label of T is not minimal. $\square$

A trail W can visit a vertex v many times, so it can be decomposed 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 $\mathring{v}W$ the trail vW without the vertex v.

Lemma 2 *Let v be the last vertex visited by a closed trail T among vertices non-exhausted by T, and let w be the next vertex in T. Then*

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

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

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

Lemma 3 *Let T be a closed trail exhausting r and let v be the last vertex visited by T among vertices non-exhausted by T. Suppose that T is the closed trail of minimum label among all closed trails exhausting $\mathring{v}T$.*

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

PROOF. T is the closed trail of minimum label exhausting $\overset{\circ}{v}T$, and Z exhausts $\overset{\circ}{v}T$, hence $l(Z) \geq l(T)$, in particular $l(Z) \geq l(Tv)$. Also Z and $(Tv)W(vT)$ exhaust vT. Hence $l(Z) \leq l((Tv)W(vT))$, concluding that $Z = (Tv)Z'$.

By Lemma 2 the only way to visit vertices in $\overset{\circ}{v}T$ is using the arc vw, and $\overset{\circ}{v}T$ is the trail of minimum label exhausting $V(\overset{\circ}{v}T)$ 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 $Z'' = W$. □

We give the following algorithm to construct the minimal Eulerian trail starting at r.

Algorithm 1 Compute the minimal Eulerian trail starting at 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.

Theorem 1 *Algorithm 1 finishes with an Eulerian trail starting at r and its label is the minimal one among all Eulerian trails starting at 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 will finish in a finite number of steps.

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 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 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 1 it is the minimal closed trail exhausting $\overset{\circ}{v}{}^i T^i$.

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

Remark that this algorithm use each arc at most twice, therefore it can be implemented in $\mathcal{O}(|A(G)|)$.

3 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 such that every word in $\mathcal{D}$ (and no other n-tuple) occurs exactly once. Historically, de Bruijn sequence was studied in an arbitrary alphabet considering the language of all the n-tuples. In [10] 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.

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 will 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,

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

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 [5] are the particular case of $\mathcal{D} = N^{n+1}$.

We label 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 $P = e_0 \dots e_m$ be a trail over $G^{\mathcal{D},n}$ of length $m \geq n$. Then P finishes in a vertex u if and only if u is a suffix of $l(P) = l(e_0) \dots l(e_m)$. This property explain 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 M be the vertex of minimum label among all vertices. Clearly, the minimal de Bruijn sequence has M as prefix. Hence, the minimal Eulerian trail over $G^{\mathcal{D},n}$ start at an (unknown) vertex and after n steps it arrives to M. Therefore if we start our Algorithm 1 in the vertex M we obtain the Eulerian trail of minimal label starting at M which have label $B = B' \cdot M$. Hence $M \cdot B'$ is the minimal de Bruijn sequence of span $n+1$ for $\mathcal{D}$.

References

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

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

[3] P. A. Pevzner, L-tuple dna sequencing: computer analysis, J. Biomol. Struct. Dyn. 7 (1989) 63–73.

[4] P. A. Pevzner, H. Tang, M. S. Waterman, An eulerian path approach to dna fragment assembly, Proceedings of the National Academy of Sciences 98 (17) (2001) 9748–9753.

[5] N. G. de Bruijn, A combinatorial problem, Nederl. Akad. Wetensch., Proc. 49 (1946) 758–764.

[6] S. K. Stein, The mathematician as an explorer, Sci. Amer. 204 (5) (1961) 148–158.

[7] J.-C. Bermond, R. W. Dawes, F. Ö. Ergincan, De Bruijn and Kautz bus networks, Networks 30 (3) (1997) 205–218.

[8] F. Chung, P. Diaconis, R. Graham, Universal cycles for combinatorial structures, Discrete Math. 110 (1-3) (1992) 43–59.

[9] W. T. Tutte, Graph theory, Vol. 21 of Encyclopedia of Mathematics and its Applications, Addison-Wesley Publishing Company Advanced Book Program, Reading, MA, 1984.

[10] E. Moreno, Lyndon words and de Bruijn sequences in a subshift of finite type, in: T. Harju, J. Karhumäki (Eds.), Proceedings of WORDS'03, no. 27 in TUCS General Publications, Turku Centre for Computer Science, Turku, Finland, 2003, pp. 400–410.