MINIMAL EULERIAN TRAIL IN A LABELED DIGRAPH

MARTÍN MATAMALA AND EDUARDO MORENO

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.

1. 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 [?]).

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 [?]) 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 [?, ?]).

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 [?]). 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

1991 Mathematics Subject Classification. Primary: 05C45; Secondary: 05C20.
Key words and phrases. Eulerian graphs, labeled digraph, de Bruijn sequence.

Partially supported by ECOS C00E03 (French-Chilean Cooperation), Programa Iniciativa Científica Milenio P01-005, and CONICYT Ph.D. Fellowship.
sequences" and were originally studied in [?] by N. G. De Bruijn for the binary alphabet. These sequences have many different applications, such as memory wheels in computers and other technological devices, network models, DNA algorithms, pseudo-random number generation and modern public-key cryptographic schemes, to mention a few (see [?, ?, ?]). More details about this application are discussed in Section 3.

By the BEST theorem (see [?]), 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 \((\gamma - 1)!\) where \(V\) is the set of vertices and \(\gamma\) 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.

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 arcs going out from the same vertex have different labels.

A trail is an alternating sequence \(W = v_1a_1v_2a_2\ldots v_{k-1}a_{k-1}v_k\) 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 as to define an order in which vertices are visited, which allows 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 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. \(\square\)
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 $\hat{v}W$ the trail $vW$ without the first vertex $v$ but containing the first arc of $vW$.

**Lemma 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

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

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

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 $\hat{v}T$. We find the trail of minimal label exhausting all vertices in $\hat{v}T$. If $v \in S(T)$ then by Lemma 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 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 $\hat{v}T$.

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 $\hat{v}T$ and $\hat{v}T \subset S(Z)$, 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 $Z = (Tv)Z'$. \qed

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

**Theorem 1.** Algorithm 1 finishes with an Eulerian trail starting in $r$ and its label is the minimal one among all Eulerian trails starting in $r$.  

Algorithm 1 Compute the minimal Eulerian trail starting in \( r \)

\[
T \leftarrow \emptyset \\
v \leftarrow \text{NoEx}(T) \quad \{ v = r \} \\
\text{while } v \neq \text{NULL} \text{ do} \\
\quad W \leftarrow W(G \setminus A(T), v).\overline{} \quad \text{over } G \setminus A(T). \\
\quad T \leftarrow (Tv)W(vT) \\
\quad v \leftarrow \text{NoEx}(T). \\
\text{end while}
\]

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

**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.

We define inductively \( G^i = G \setminus A(T^{i-1}), v^i = \text{NoEx}(T_i), W^i = W(G^i, v^i) \) 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 \( v^i T^i \). 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 \( v^1 T^1 \). Let \( T^{i-1} \) be the closed trail of minimum label exhausting \( v^{i-1}T^{i-1} \). 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 \( v^i T^i \).

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. \( \square \)

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 \( \mathcal{O}(|A(G)|) \), 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 1, the minimal de Bruijn sequence starting at \( u \) is 001122 but starting at \( v \) is 100122.

### 3. An application: minimal de Bruijn sequence

Given a set \( D \) of words of length \( n \), a de Bruijn sequence of span \( n \) is a periodic sequence \( B \) such that every word in \( D \) (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 [?] 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 [?] 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 $\varepsilon$), then $p$ is called a prefix of $w$, and if $v$ is empty then is called a suffix of $w$.

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

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

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

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

and arc set

$A(G_{D,n}) = \{(\alpha u, v \beta) | \alpha, \beta \in N, \alpha u \beta \in D\}$

Note that the original definitions of de Bruijn sequences and de Bruijn graph given in [?] are the particular case of $D = N^{n+1}$.

We label the arcs of the graph $G_{D,n}$ 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 $G_{D,n}$ 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: $B_{D,n+1}$ is the label of an Eulerian trail of $G_{D,n}$. Therefore, given a dictionary $D$, the existence of a de Bruijn sequence of span $n + 1$ is characterized by the existence of an Eulerian trail over $G_{D,n}$.

Let $D$ be a dictionary such that $G_{D,n}$ 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 $G_{D,n}$ starts at an (unknown) vertex and after $n$ steps it arrives to $z$. Therefore if we start our Algorithm 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 \( D \).

E-mail address: mmatamal@dim.uchile.cl

E-mail address: emoreno@dim.uchile.cl

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.